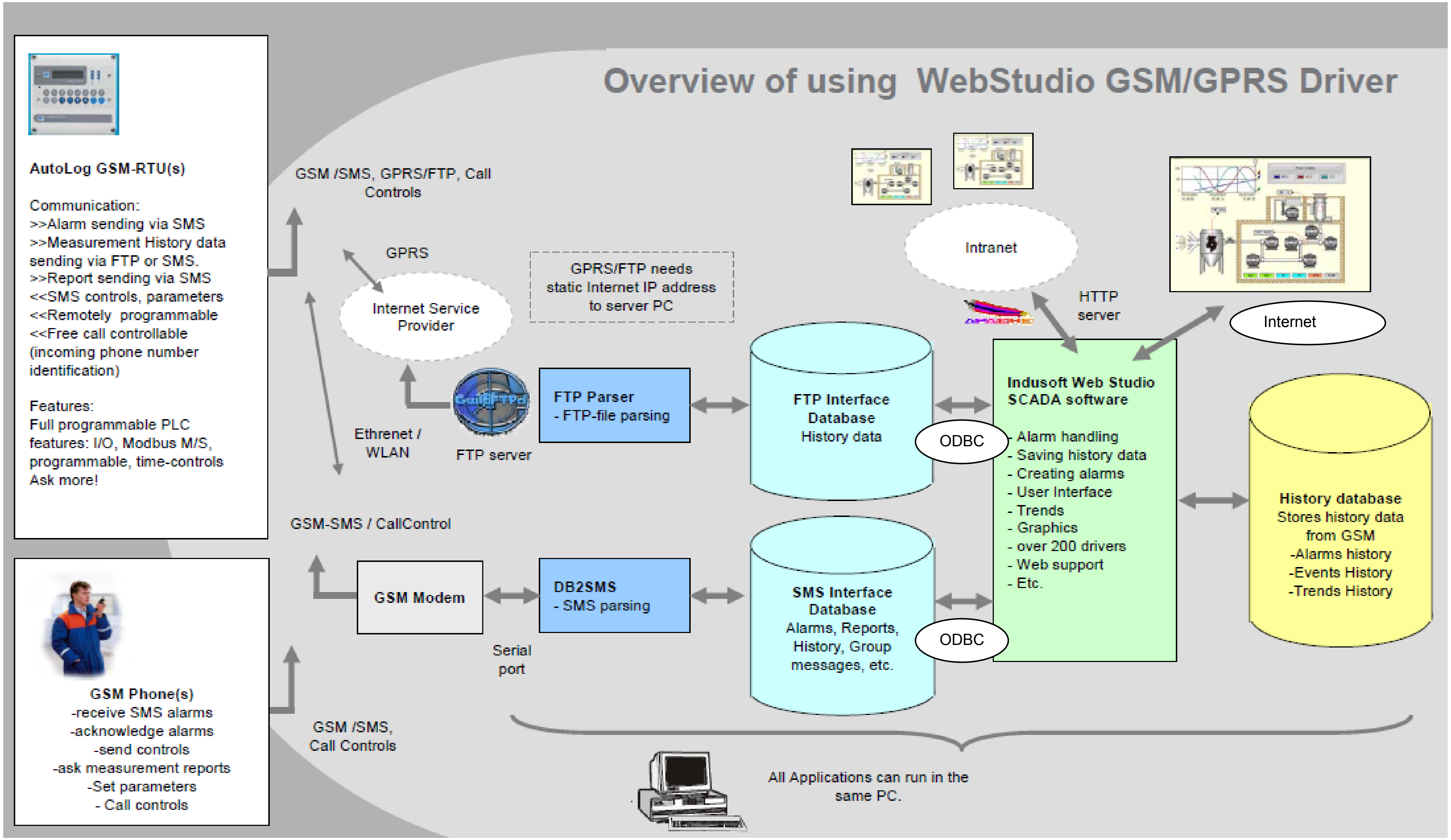
FF-AUTOMATION

User Manual

v.2.04g

- AutoLog GSM-PLC (GPRS & SMS) drivers for
- Indusoft Web Studio

Updated 15th April 2015



Overview of using WebStudio GSM/GPRS Driver

AutoLog GSM-RTU(s)

Communication:
>>Alarm sending via SMS
>>Measurement History data sending via FTP or SMS.
>>Report sending via SMS
<<SMS controls, parameters
<<Remotely programmable
<<Free call controllable (incoming phone number identification)

Features:
Full programmable PLC features: I/O, Modbus M/S, programmable, time-controls
Ask more!

GSM Phone(s)
-receive SMS alarms
-acknowledge alarms
-send controls
-ask measurement reports
-Set parameters
- Call controls

GSM /SMS, GPRS/FTP, Call Controls

GPRS

Internet Service Provider

GPRS/FTP needs static Internet IP address to server PC

Ethernet / WLAN

FTP server

FTP Parser
- FTP-file parsing

FTP Interface Database
History data

ODBC

Intranet

HTTP server

Internet

Indusoft Web Studio SCADA software
- Alarm handling
- Saving history data
- Creating alarms
- User Interface
- Trends
- Graphics
- over 200 drivers
- Web support
- Etc.

History database
Stores history data from GSM
-Alarms history
-Events History
-Trends History

GSM-SMS / CallControl

GSM Modem

Serial port

DB2SMS
- SMS parsing

SMS Interface Database
Alarms, Reports, History, Group messages, etc.

ODBC

GSM /SMS, Call Controls

All Applications can run in the same PC.

1 Drivers description

1.1 DB2SMS -driver

DB2SMS is the name of the Web Studio SCADA’s GSM-SMS driver. DB2SMS driver can receive and send SMS messages through GSM modem with FF-Automation’s AutoLog GSM-RTU devices. It can be used also for SMS alarms forwarding from Indusoft Web Studio. Driver uses PostGreSQL database as an interface. Driver is made by FF-Automation so it is not included in the standard Indusoft software package.

1.2 FTP_Parser -driver

FTP_Parser is the name of the Web Studio SCADA’s GPRS/FTP file parser. It can be used to parse the incoming FTP files (send by AutoLog GSM-RTU using GPRS communication). Driver uses PostGreSQL database as an interface. Driver is made by FF-Automation so it is not included in the standard Indusoft software package.

1.3 Tested versions

DB2SMS v. 1.3.4456.18311

- Tested with Windows XP
- Tested with Windows 7 (32-bit)
- Tested with Windows 7 (64 bit)
- postgresql-8.4.16-1-windows.exe

FTP_Parser v. 1.0.0.0

- Tested with Windows XP
- Tested with Windows 7 (32-bit)
- postgresql-8.4.16-1-windows.exe

2 Contents

3 Preparations

3.1 Manual Description

This manual serves as installation and configuration manual for setting up the interface between AutoLog GSM-RTUs and Indusoft Web Studio SCADA software.

1. Needed components.
2. Step-by-step installation manual for installing all the needed software components.
3. Configuring the software and hardware components.
4. Part of this manual shows how to use and test the system
5. Index part is showing all the example configurations, codes and interface descriptions
6. Needed files

(*) Contact FF-Automation to get the permission to the FTP server to download the latest versions of the manuals and Demo applications!

3.2 Needed components

3.2.1 Needed hardware components

- Windows PC
- Suggested Operating System: Windows 7 Pro
- Serial ports: 1 minimum, 2 suggested
- Ethernet port: 1 minimum
- Display: 24” or bigger is suggested for SCADA, but smaller can be used too.
- 8GB or more RAM suggested for SCADA
- 500 GB hard disk suggested, 2 hard disk RAID system is suggested for SCADA

Normal “known reliable brand like HP” desktop PC woks ok, but for 24/7/365 SCADA application it is suggested to have server like reliability. FF-Automation can suggest suitable PC for your application.

3.2.2 Needed hardware accessories:

- AutoLog GSM-PLC(s), GSM-4, GSM-8, GSM-16 or GSM-20. Also GSM-PLC demo kit is available for training and testing your application. Order from FF-Automation.
- GSM-PLC programming cable, AL Programming Cable PC - AL (RJ45), Length 2.5m, order code 903195.
- GSM modem Unit for PC (WAVECOM FASTRACK XTEND GSM/GPRS modem, stubby antenna, power, enclosure and PC to modem cable), can be ordered from FF-Automation.
- SIM cards for GSM-PLC and PC’s GSM modem. Order these from your local operator.
- Small screw driver and about 2m red and blue I/O testing wires.

3.2.3 Needed software components

- DB2SMS driver (Order from FF-Automation)
- FTP_parser driver (Order from FF-Automation)

- PostGreSQL database (Download from Internet or ask FF-Automation) http://www.postgresql.org/ This manual is written using PostgreSQL version postgresql-8.4.16-1-windows.exe. **For Window 7 (x64) you should use postgresql-9.3.6-1-windows-x64 (especially if you are using ftp_parser)**
- ODBC connectors. You can download ODBC connector for PostgreSQL from this site: http://www.postgresql.org/ftp/odbc/versions/msi/ Tested version psqlodbc_08_04_0200.zip. **For Window 7 (x64) you should use psqlodbc_09_03_0300-x64-1 (especially if you are using ftp_parser)**
- Filezilla FTP server.
  FileZilla_Server-0_9_49.exe (used in this manual)
  Download free Filezilla FTP server from https://filezilla-project.org/
- Indusoft Web Studio SCADA software. Download latest version of Indusoft Web Studio from Indusoft's web page: www.indusoft.com (You need to register to the page) or ask FF-Automation. Licenses can be ordered from FF-Automation. FF-Automation can also give training.
- GsmProgrammer software for programming AutoLog GSM-PLCs. The latest version while writing this manual is v. 1.6.512.2. (Order from FF-Automation)

(*) Contact FF-Automation and order "GSM_PLC_Driver_for_Indusoft_Web_Studio" and you will get DVDs, which contain all the needed software components and manuals.

3.2.4 Needed Manuals

- This manual!
- AutoLog GSM PLC User Manual. (ask FF-Automation)
- Indusoft Web Studio Manuals (download from www.indusoft.com)
- Indusoft Web Studio training videos (check from www.indusoft.com)

(*) Contact FF-Automation to get the permission to the FTP server to download the latest versions of the manuals and Demo applications!

3.2.5 Needed demo application and accessory files

- Smsindusoft_2013.backup –file (for DB2SMS driver configuration)
- GSM_to_Induosft_FTP_DEMO –zipped project file (for Indusoft Web Studio FTP_parser demo project)

3.3 Application types and driver usage

3.3.1 Indusoft Web Studio (IWS) SMS Alarms forwarding to GSM phones

Requirements:

- GSM modem and accessories
- DB2SMS driver
- PostGreSQL database
- ODBC
- Indusoft Web Studio SCADA software

Needed files

This is the simplest application type. You need to install PostGreSQL but you can skip the History database creation.

3.3.2 IWS bi-directional SMS communication with AutoLog GSM-RTUs

Requirements:

- GSM modem and accessories
- DB2SMS driver

- AutoLog GSM-RTU(s)
- PostGreSQL database
- ODBC
- Indusoft Web Studio SCADA software
- AutoLog GsmProgrammer

If you need to read different kinds of data from GSM-RTUs and also to send control commands. You can read measurements data but if you need to read lots of measurement data it might be more cost effective to use GPRS/FTP driver. Anyway you can use both DB2SMS driver and FTP_file_parser driver at the same time.

3.3.3 IWS GPRS/FTP data reading from AutoLog GSM-RTUs

Requirements:

- AutoLog GSM-RTU(s)
- FTP_parser -driver
- Static Internet IP address (If dynamic IP address changes the GSM-PLC tries to send to FTP files to wrong IP address.)
- Public IP address (or defined FTP port (normally port 21) forward for router "NAT setup")
- PostGreSQL database
- ODBC
- Indusoft Web Studio SCADA software
- AutoLog GsmProgrammer

If you need only to read measurement data from GSM-RTUs, you can use FTP_parser and you don't need GSM modem, because FTP files are send from GSM-PLC's to your PC using GPRS / Internet.

See also FTP_Demo – Indusoft Web Studio Demo application and FTP_Demo GsmProgrammer application.

3.3.4 IWS GPRS/FTP data reading and SMS communication with AutoLog GSM-RTUs

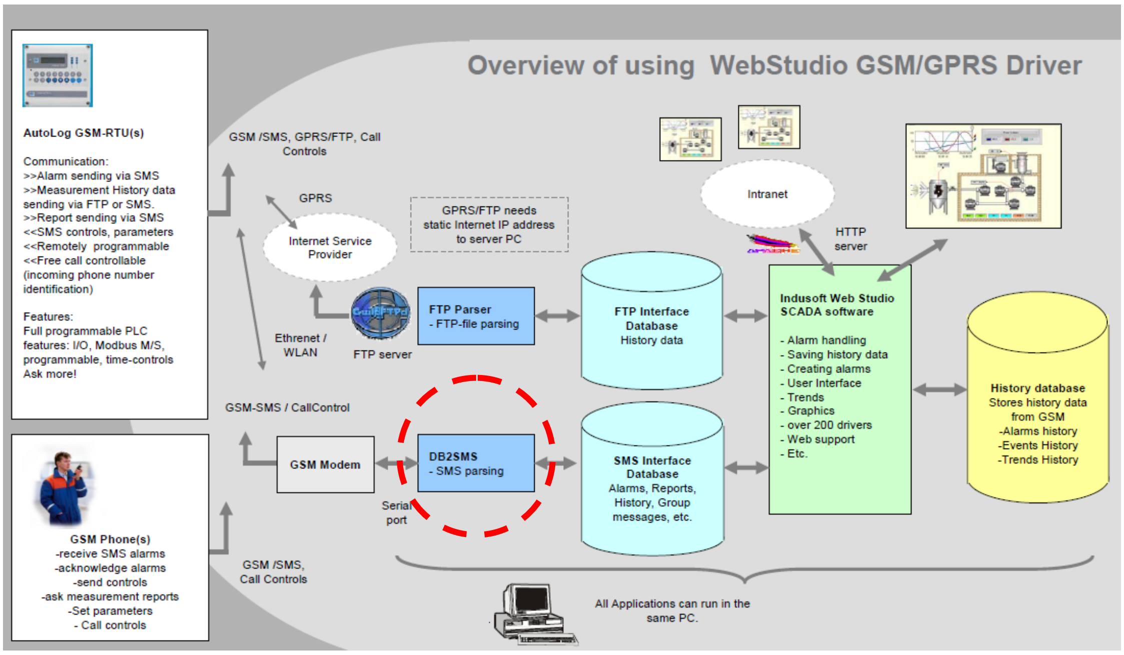
- GSM modem and accessories
- AutoLog GSM-RTU(s)
- DB2SMS driver
- FTP_parser -driver
- Static Internet IP address (If dynamic IP address changes the GSM-PLC tries to send to FTP files to wrong IP address.)
- Public IP address (or defined FTP port (normally port 21) forward for router "NAT setup")
- PostGreSQL database
- ODBC
- Indusoft Web Studio SCADA software
- AutoLog GsmProgrammer

FTP_parser allows only one directional communication between SCADA PC<-GSM-RTUs, so you cannot for example send control commands from SCADA PC to GSM-RTUs. Common usage is to use both DB2SMS driver with GSM modem for sending controls and receiving alarms and FTP_parser for collecting measurements data.

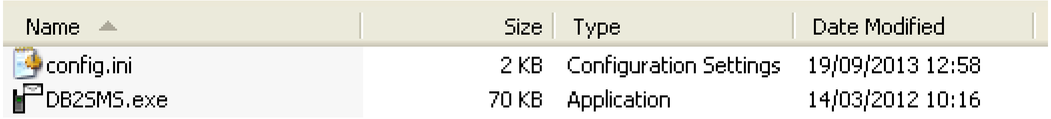
4 Installing the software components

4.1 Installing DB2SMS –driver

If you are using only the GSRS/FTP communication, you don't need to install DB2SMS driver.



1. Check that you have installed MS .Net Framework 2.0 (This can be checked form control panel Add/remove programs list) http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en
   Windows 7 doesn't need this!

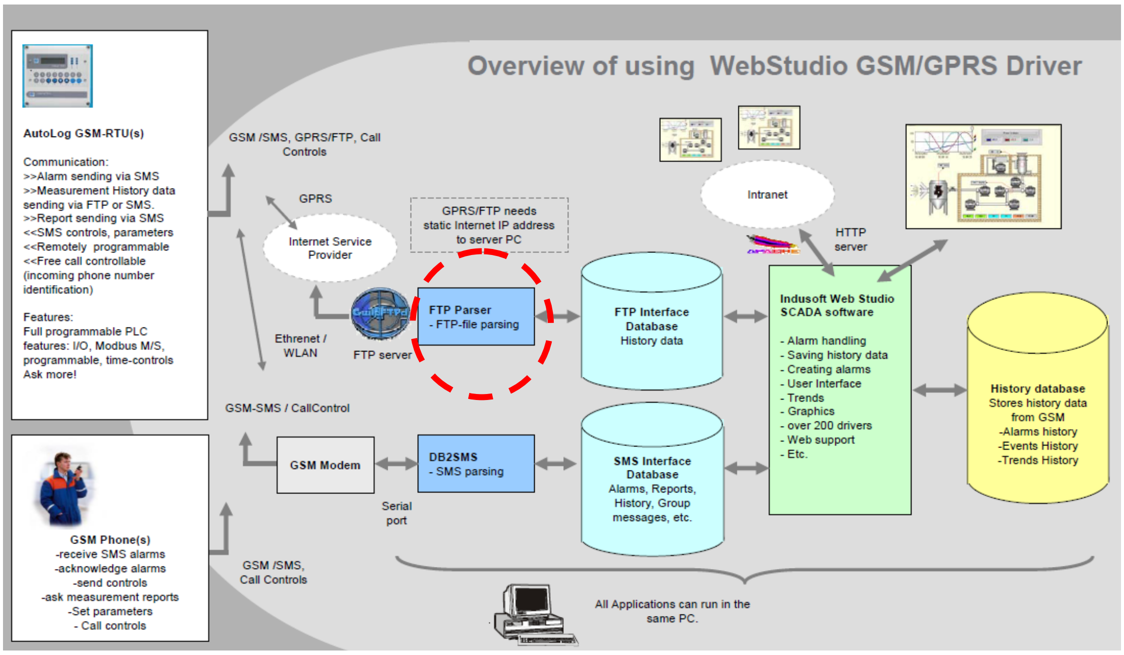


Name	Size	Type	Date Modified
config.ini	2 KB	Configuration Settings	19/09/2013 12:58
DB2SMS.exe	70 KB	Application	14/03/2012 10:16

2. Copy DB2SMS program (DB2SMS.exe) and configuration file (config.ini) to your PC. (e.g. C:\DB2SMS directory) or as suggested under Indusoft Web Studio Projects directory (e.g. c:\IWS_Projects\IWS_Project_name\DB2SMS\.
3. DB2SMS cannot be used without PosGreSQL database or other components described in this manual, Look "Installing PostGreSQL database chapter".
4. Installation is ready

4.2 Installing FTP_parser –driver

If you are using only the GSM-SMS communication, you don’t need to install FTP_parser driver.



Overview of using WebStudio GSM/GPRS Driver

AutoLog GSM-RTU(s)

Communication:
>>Alarm sending via SMS
>>Measurement History data sending via FTP or SMS.
>>Report sending via SMS
<<SMS controls, parameters
<<Remotely programmable
<<Free call controllable (incoming phone number identification)

Features:
Full programmable PLC features: I/O, Modbus M/S, programmable, time-controls
Ask more!

GSM Phone(s)
-receive SMS alarms
-acknowledge alarms
-send controls
-ask measurement reports
-Set parameters
- Call controls

GSM /SMS, GPRS/FTP, Call Controls

GPRS

Internet Service Provider

GPRS/FTP needs static Internet IP address to server PC

Ethrenet / WLAN

FTP server

FTP Parser
- FTP-file parsing

GSM-SMS / CallControl

GSM Modem

Serial port

DB2SMS
- SMS parsing

GSM /SMS, Call Controls

FTP Interface Database
History data

SMS Interface Database
Alarms, Reports, History, Group messages, etc.

Intranet

HTTP server

Indusoft Web Studio SCADA software

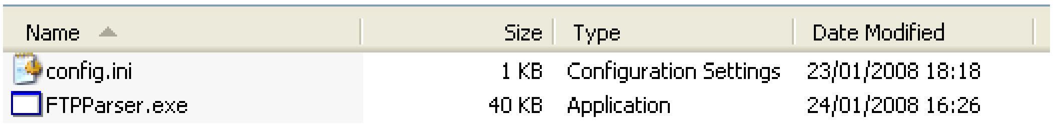
- Alarm handling
- Saving history data
- Creating alarms
- User Interface
- Trends
- Graphics
- over 200 drivers
- Web support
- Etc.

History database
Stores history data from GSM
-Alarms history
-Events History
-Trends History

All Applications can run in the same PC.

Installation of FTP History File Parser is quite a simple process:

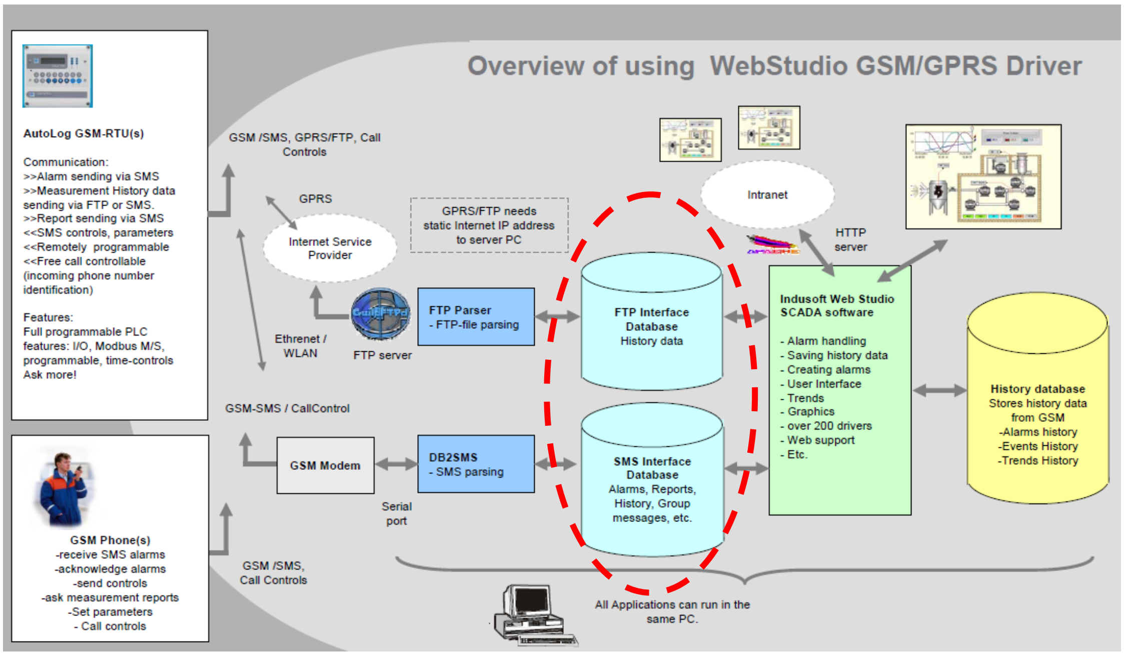
1. Install Microsoft .NET Framework 2.0 – not needed for Windows 7
2. Copy FTP_parser program (FTPParser.exe) and configuration file (config.ini) to your PC. (e.g. C:\FTP_parser\ directory) or as suggested under Indusoft Web Studio Projects directory (e.g. c:\IWS_Projects\IWS_Project_name\FTP_Parser\.
3. FTP_Parser cannot be used without PosGreSQL database or FTP server or other components described in this manual, Look “Installing PostGreSQL database chapter”.
4. Installation is ready



Name	Size	Type	Date Modified
config.ini	1 KB	Configuration Settings	23/01/2008 18:18
FTPParser.exe	40 KB	Application	24/01/2008 16:26

4.3 Installing PostGreSQL database

This is needed for both GPRS/FTP communication and GSM-SMS communication.



Additional information and latest versions of PostgreSQL can be found and downloaded free from the Internet.

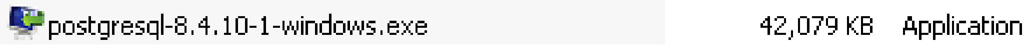
http://www.postgresql.org/

This manual is written using PostgreSQL version postgresql-8.4.16-1-windows.exe. (Some of the pictures are taken from the older version 8.4.10.1, but the procedures are the same) **For Window 7 (x64) you should use postgresql-9.3.6-1-windows-x64 (especially if you are using ftp_parser)**

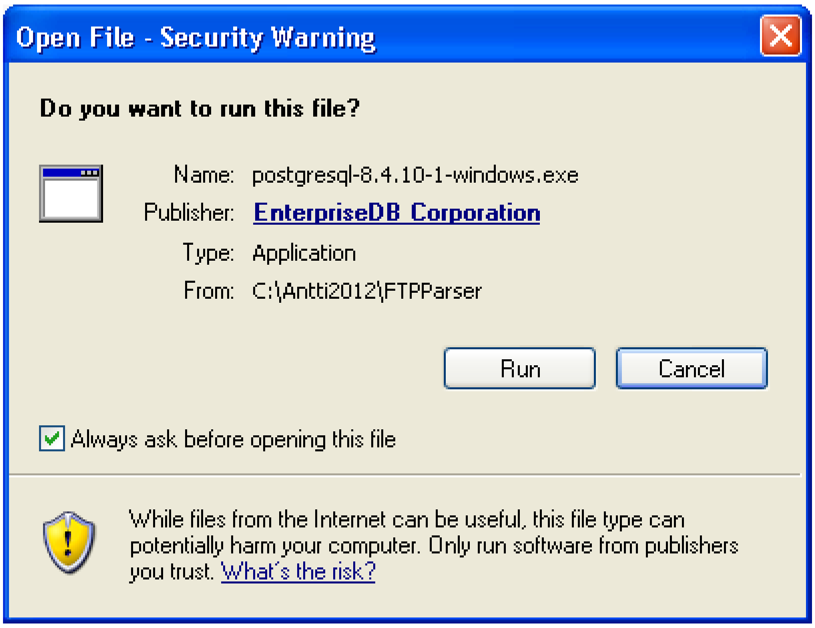
PostgreSQL database is needed both for DB2SMS driver and FTP_file_parser driver.

4.3.1 Starting installation

Copy the file to your PC's local hard disk. e.g. c:/temp/


postgresql-8.4.10-1-windows.exe 42,079 KB Application

Double click the file.



Open File - Security Warning

Do you want to run this file?

Name: postgresql-8.4.10-1-windows.exe

Publisher: EnterpriseDB Corporation

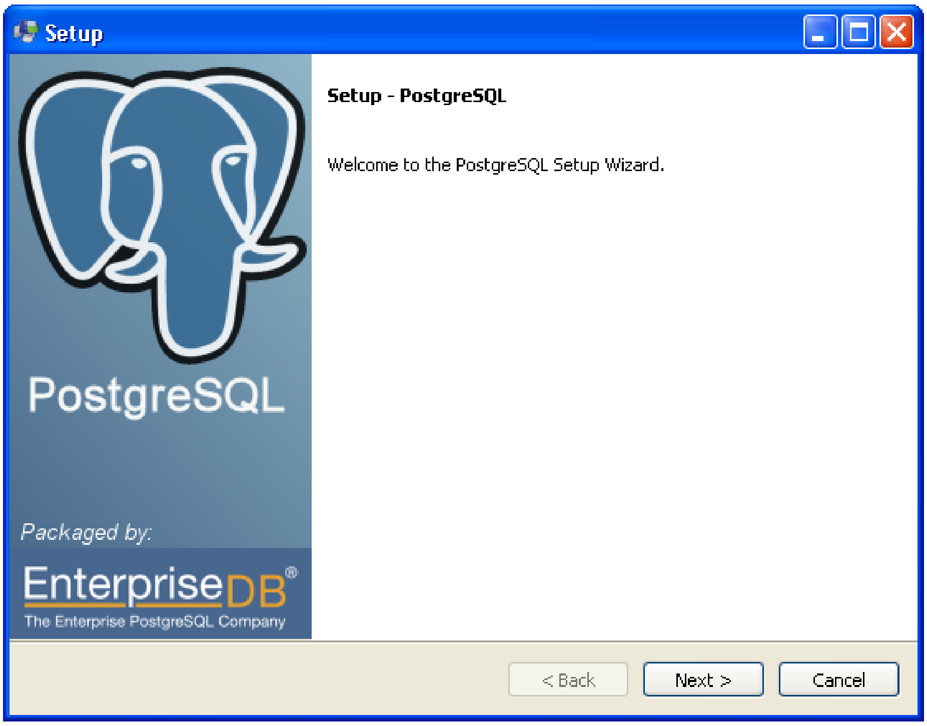
Type: Application

From: C:\Antti2012\FTPParser

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

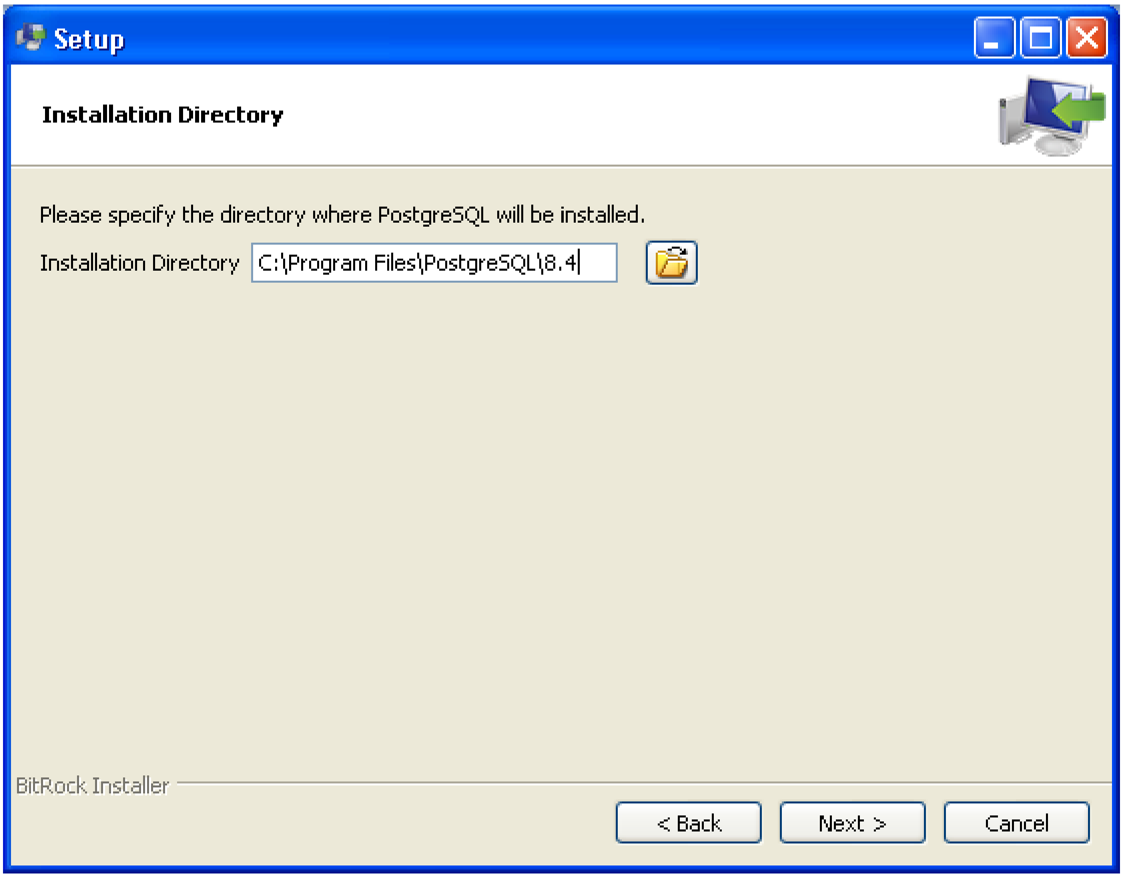
1. Click *Run*



Setup - PostgreSQL

Welcome to the PostgreSQL Setup Wizard.

2. Click Next >



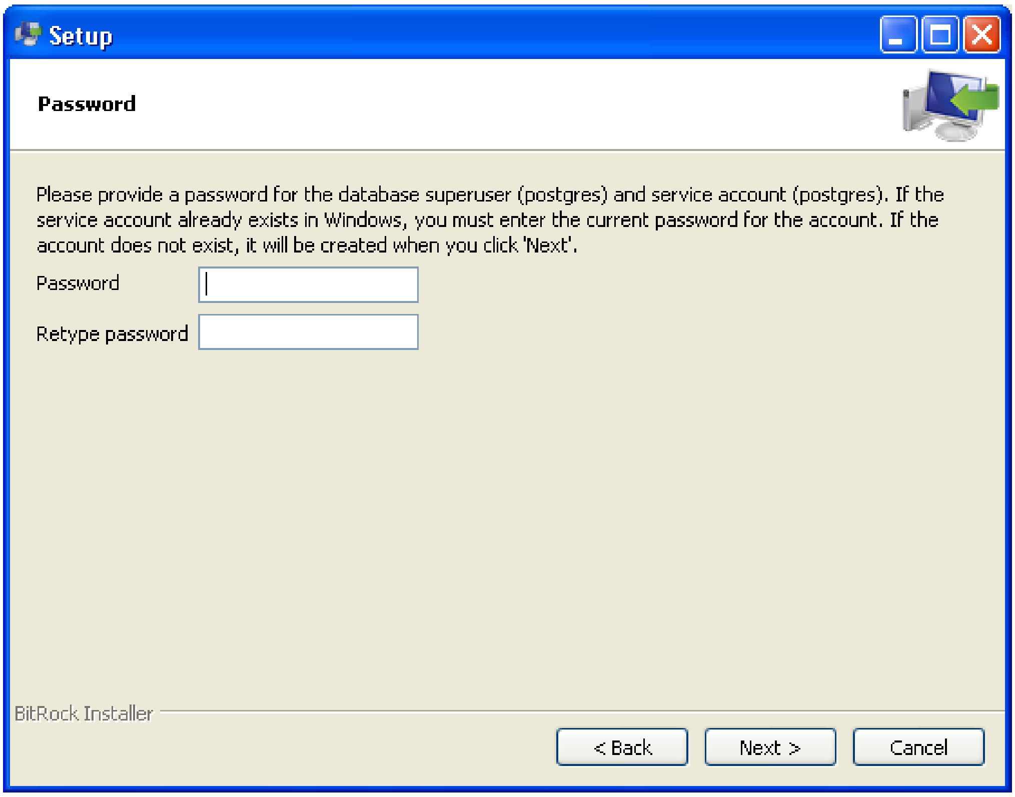
Installation Directory

Please specify the directory where PostgreSQL will be installed.

Installation Directory C:\Program Files\PostgreSQL\8.4

BitRock Installer

Leave default settings and Click Next >



Password

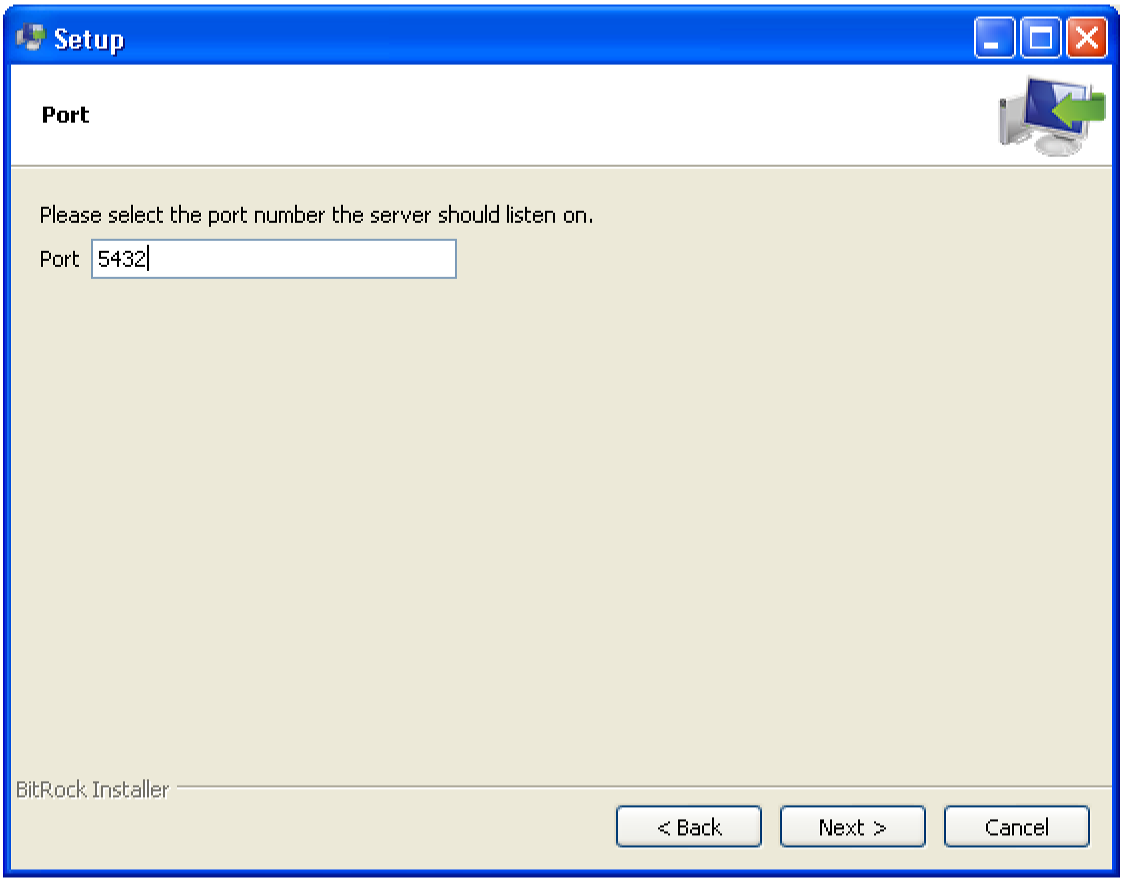
Please provide a password for the database superuser (postgres) and service account (postgres). If the service account already exists in Windows, you must enter the current password for the account. If the account does not exist, it will be created when you click 'Next'.

Password

Retype password

BitRock Installer

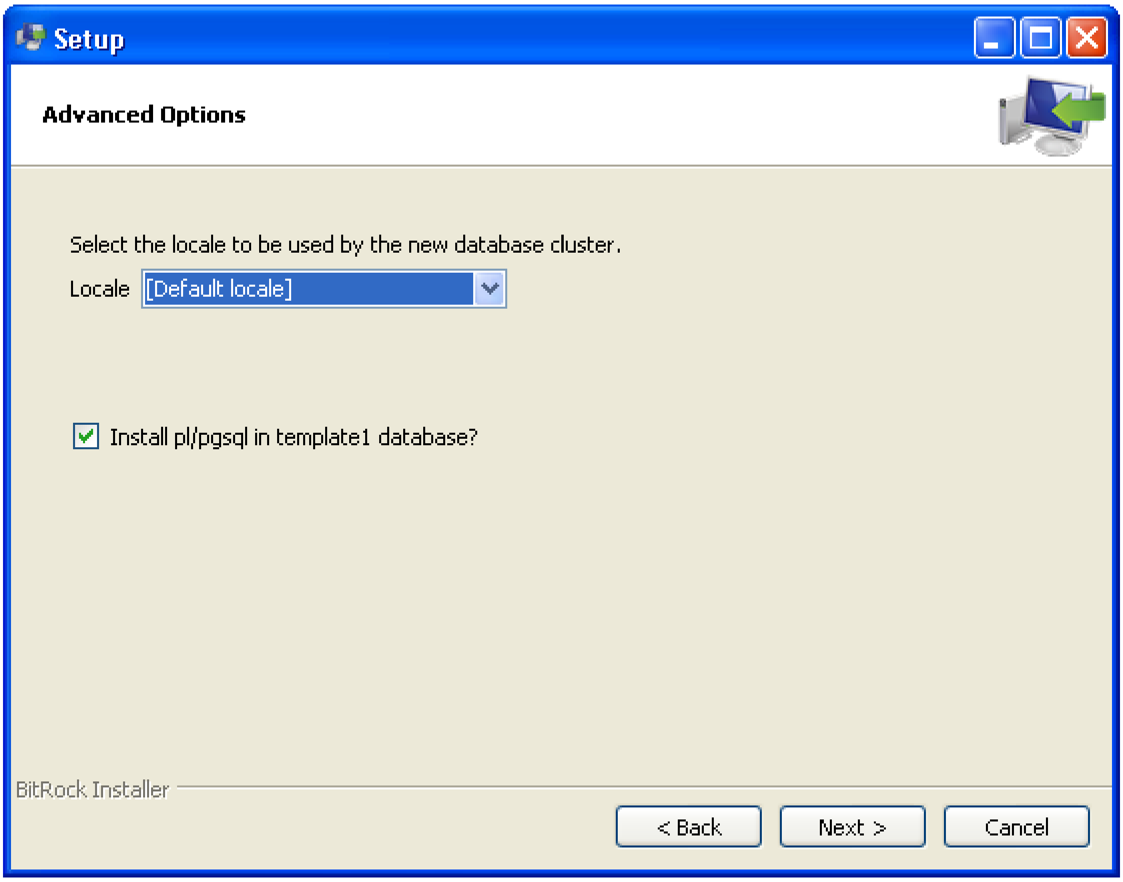
write password for superuser (REMEMBER IT AFTERWARDS!!!)



Please select the port number the server should listen on.

Port 5432

Leave default, click Next >

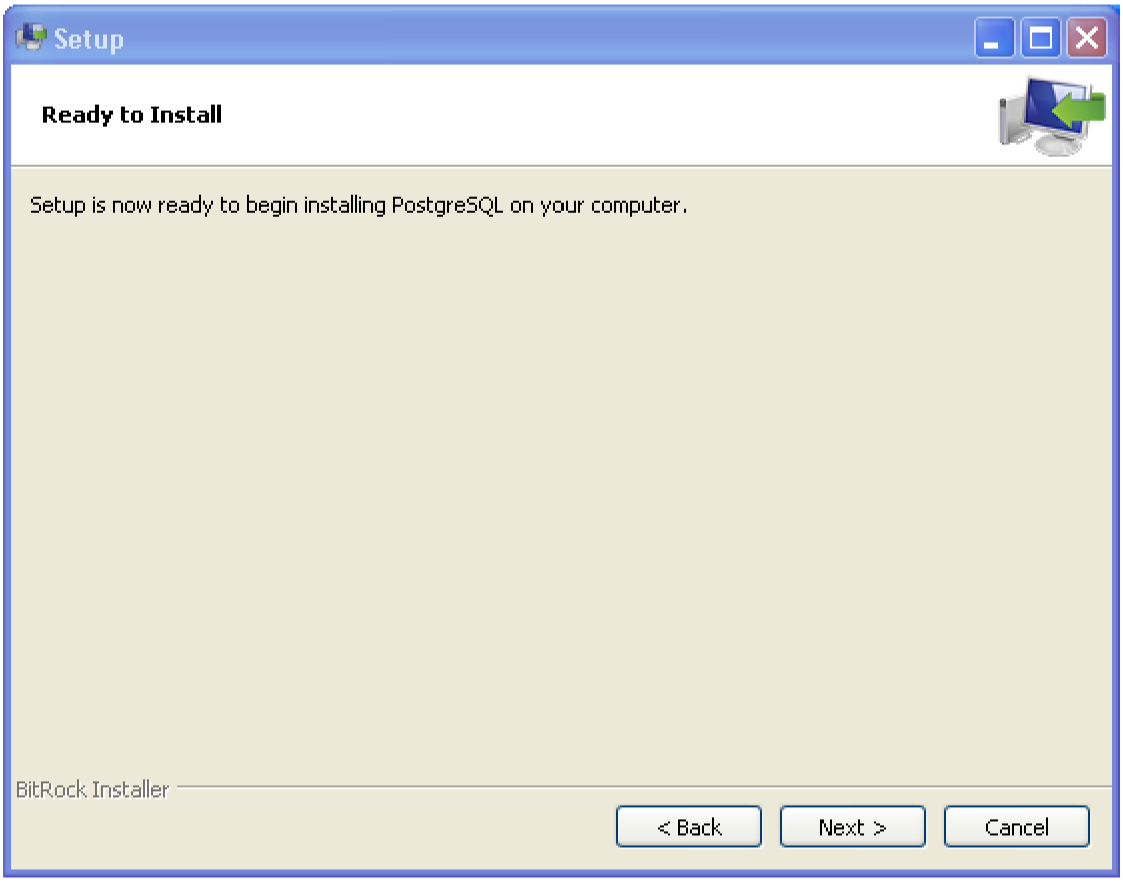


Select the locale to be used by the new database cluster.

Locale [Default locale]

☑ Install pl/pgsql in template1 database?

Leave default, click Next >



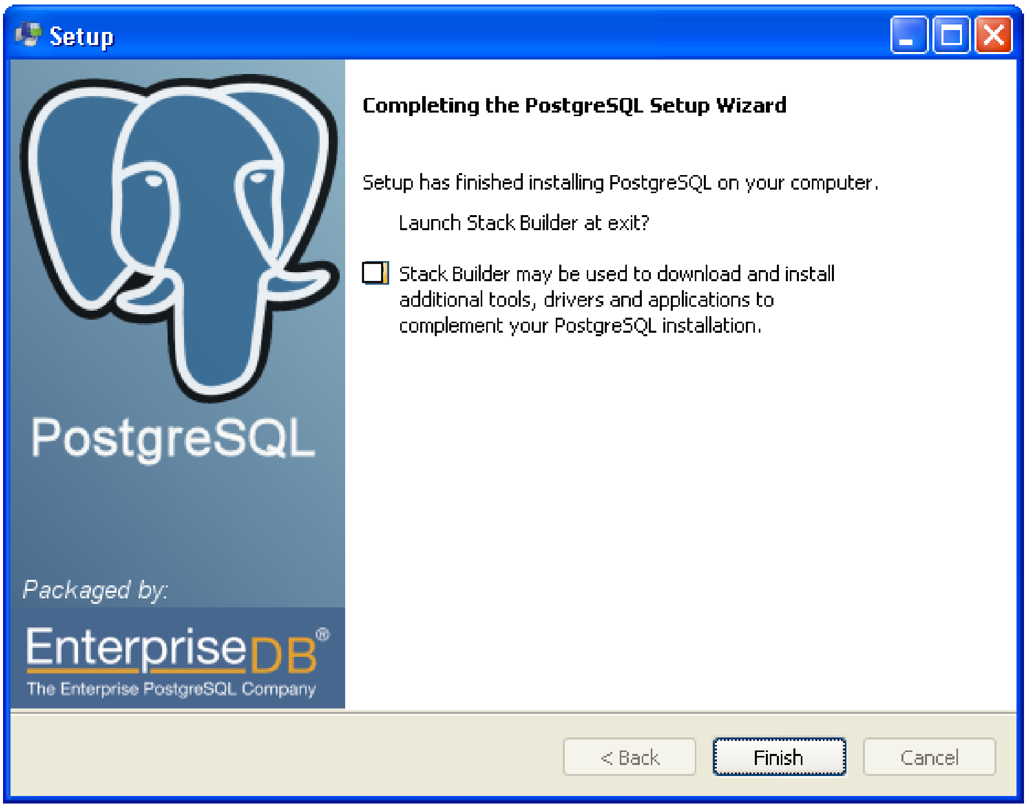
Setup

Ready to Install

Setup is now ready to begin installing PostgreSQL on your computer.

Click Next>

Wait until installation is finished.



Setup

Completing the PostgreSQL Setup Wizard

Setup has finished installing PostgreSQL on your computer.

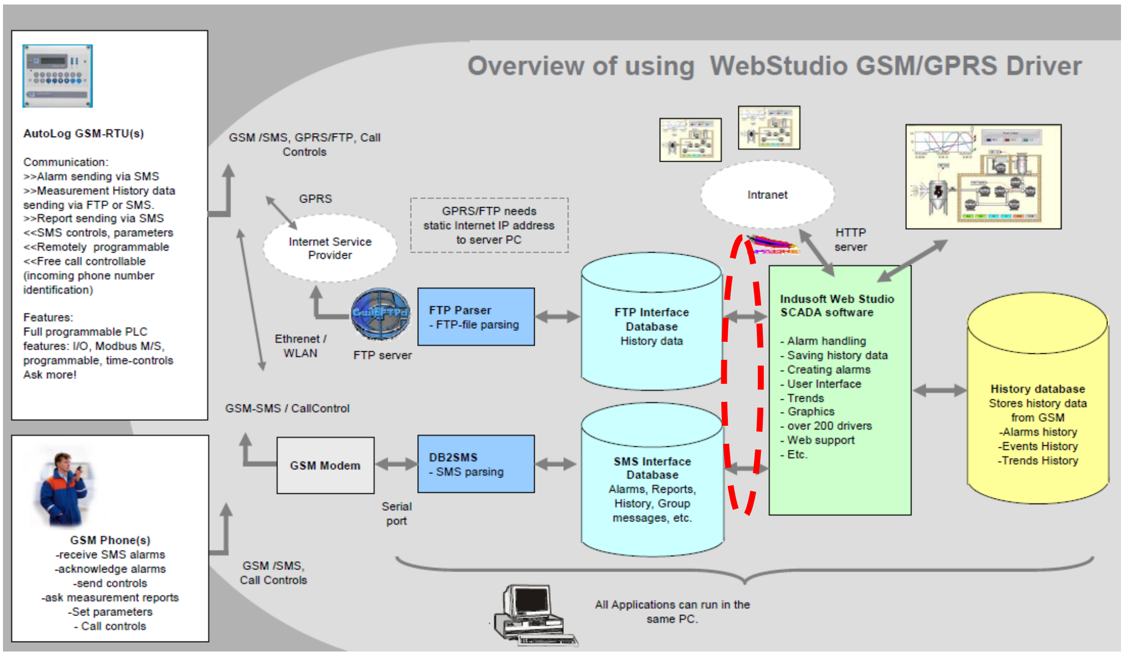
Launch Stack Builder at exit?

□ Stack Builder may be used to download and install additional tools, drivers and applications to complement your PostgreSQL installation.

Uncheck the Stack builder and Click Finish

4.4 Installing ODBC connectors for PostreSQL database

- This is needed both for DB2SMS and FTP_file_parser –drivers.



Overview of using WebStudio GSM/GPRS Driver

AutoLog GSM-RTU(s)

Communication:
>>Alarm sending via SMS
>>Measurement History data sending via FTP or SMS.
>>Report sending via SMS
<<SMS controls, parameters
<<Remotely programmable
<<Free call controllable (incoming phone number identification)

Features:
Full programmable PLC features: I/O, Modbus M/S, programmable, time-controls
Ask more!

GSM Phone(s)
-receive SMS alarms
-acknowledge alarms
-send controls
-ask measurement reports
-Set parameters
- Call controls

GPRS/FTP needs static Internet IP address to server PC

FTP Parser
- FTP-file parsing

FTP Interface Database
History data

DB2SMS
- SMS parsing

SMS Interface Database
Alarms, Reports, History, Group messages, etc.

Indusoft Web Studio SCADA software

- Alarm handling
- Saving history data
- Creating alarms
- User Interface
- Trends
- Graphics
- over 200 drivers
- Web support
- Etc.

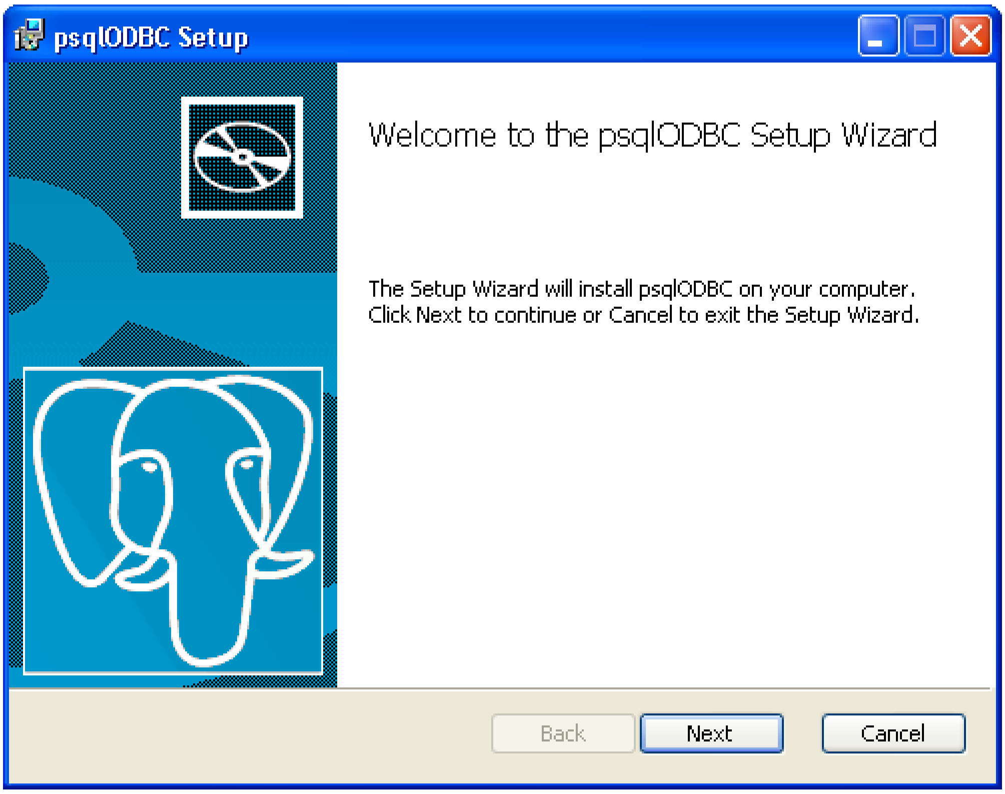
History database
Stores history data from GSM
-Alarms history
-Events History
-Trends History

All Applications can run in the same PC.

You can download ODBC connector for PostgreSQL from this site:
http://www.postgresql.org/ftp/odbc/versions/msi/
ODBC connectors are also delivered in driver CD.

Tested version psqlodbc_08_04_0200.zip
For Window 7 (x64) you should use psqlodbc_09_03_0300-x64-1 (especially if you are using ftp_parser)

Unzip this file and run setup.

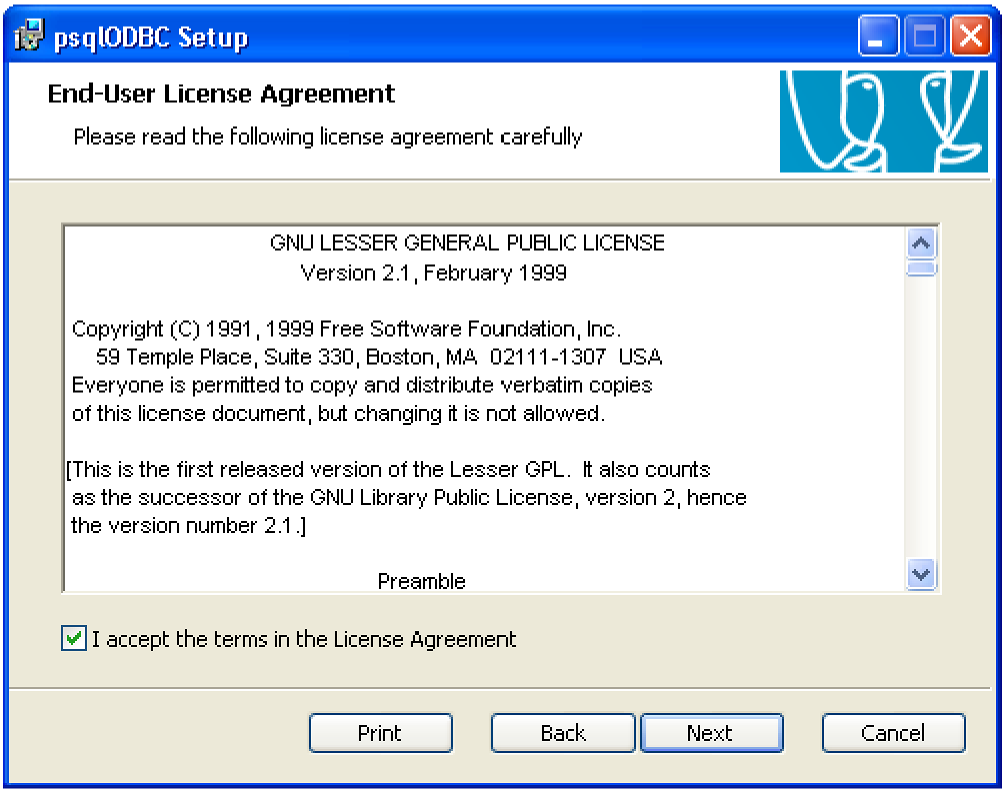


psqlODBC Setup

Welcome to the psqlODBC Setup Wizard

The Setup Wizard will install psqlODBC on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.

Click Next



psqlODBC Setup

End-User License Agreement

Please read the following license agreement carefully

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

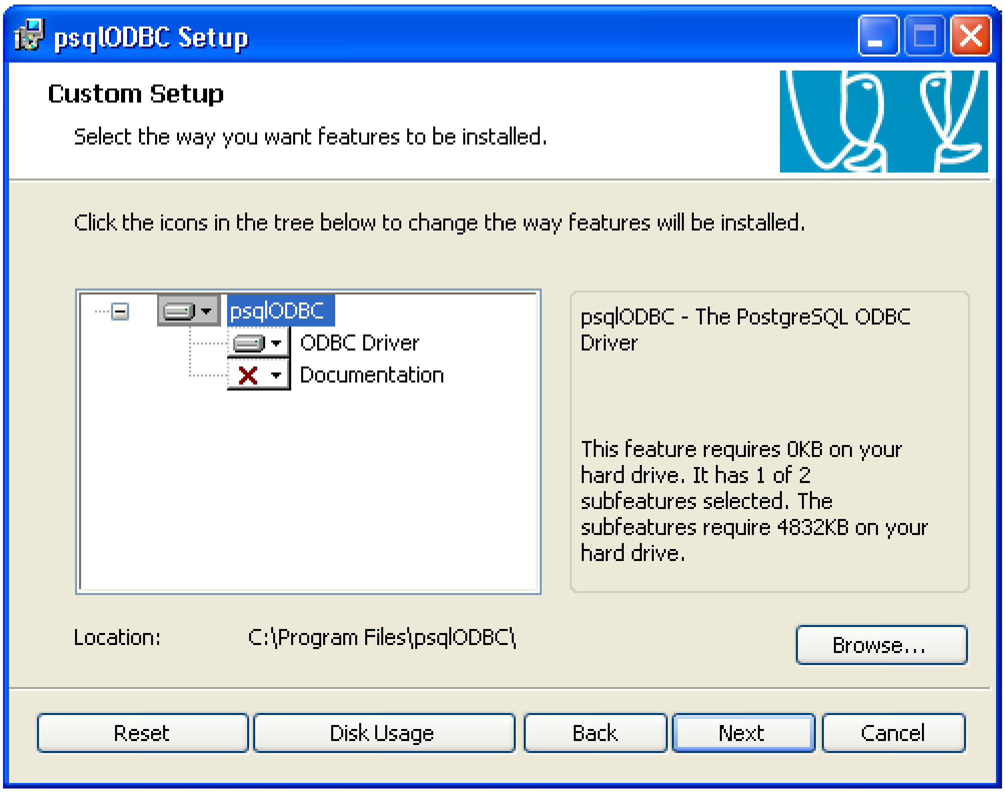
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

☑ I accept the terms in the License Agreement

Select "I accept…" Click Next



psqlODBC Setup

Custom Setup

Select the way you want features to be installed.

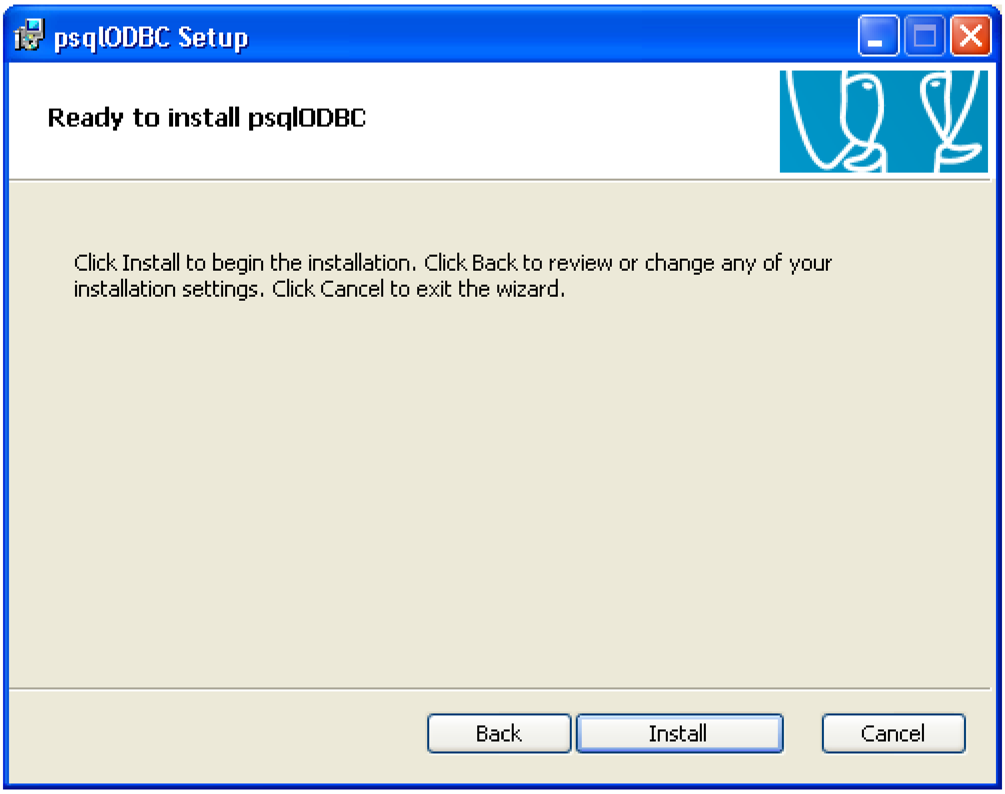
Click the icons in the tree below to change the way features will be installed.

psqlODBC - The PostgreSQL ODBC Driver

This feature requires 0KB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 4832KB on your hard drive.

Location: C:\Program Files\psqlODBC\

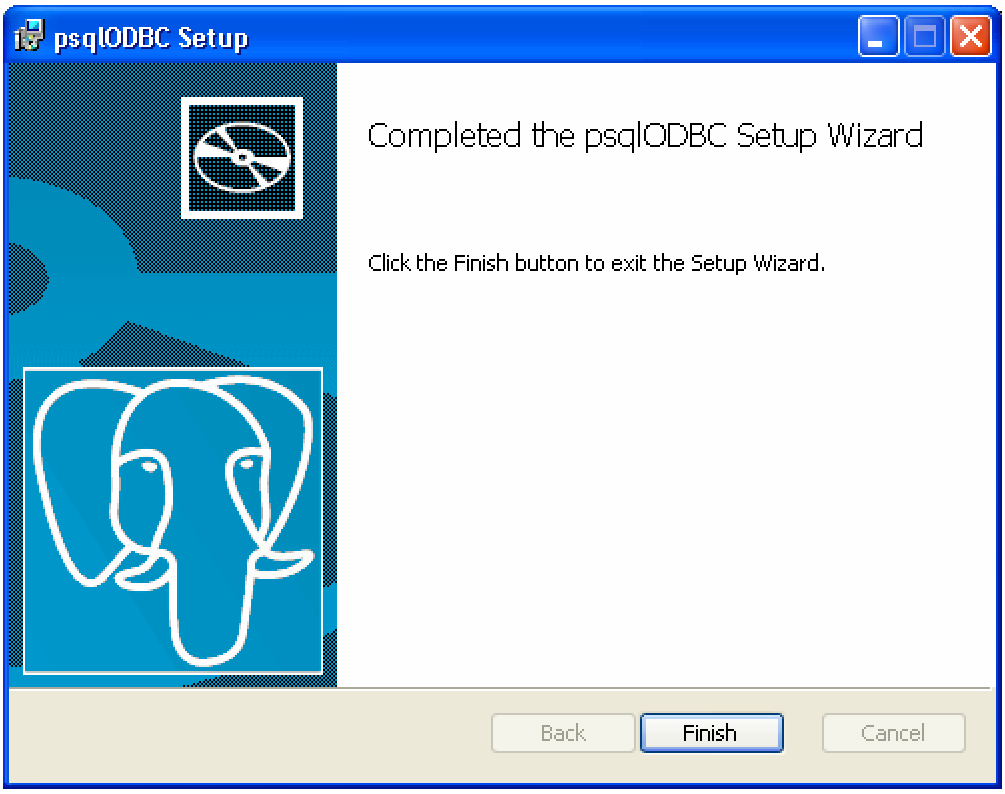
Click Next



Ready to install psqlODBC

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Click install



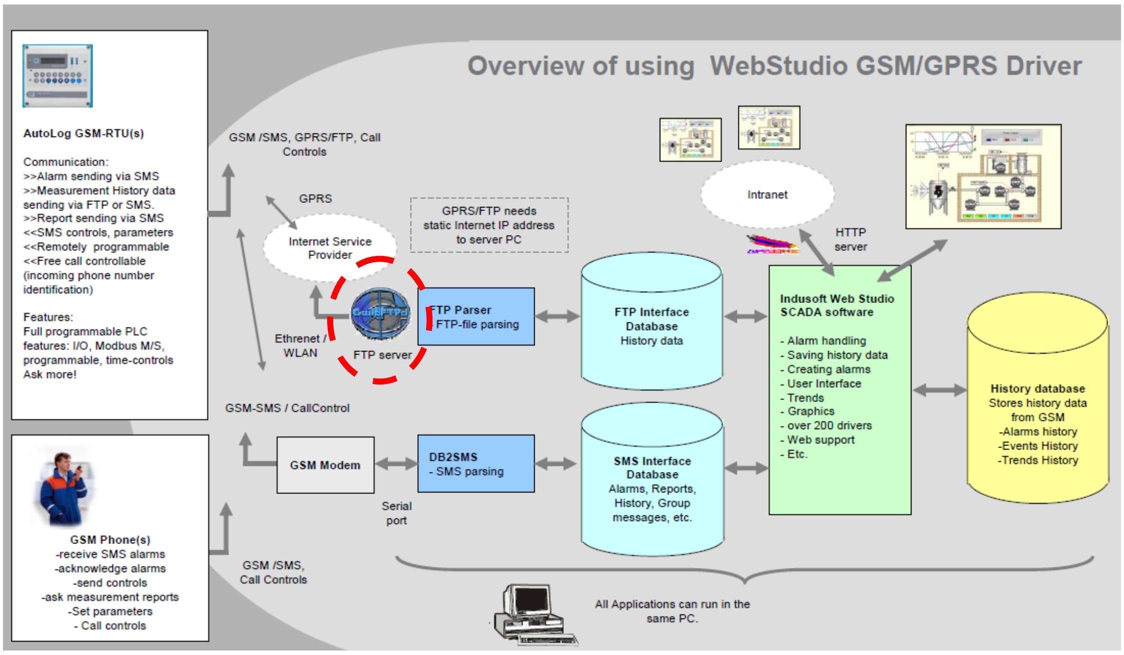
Completed the psqlODBC Setup Wizard

Click the Finish button to exit the Setup Wizard.

Click Finish

4.5 Installing FTP server software

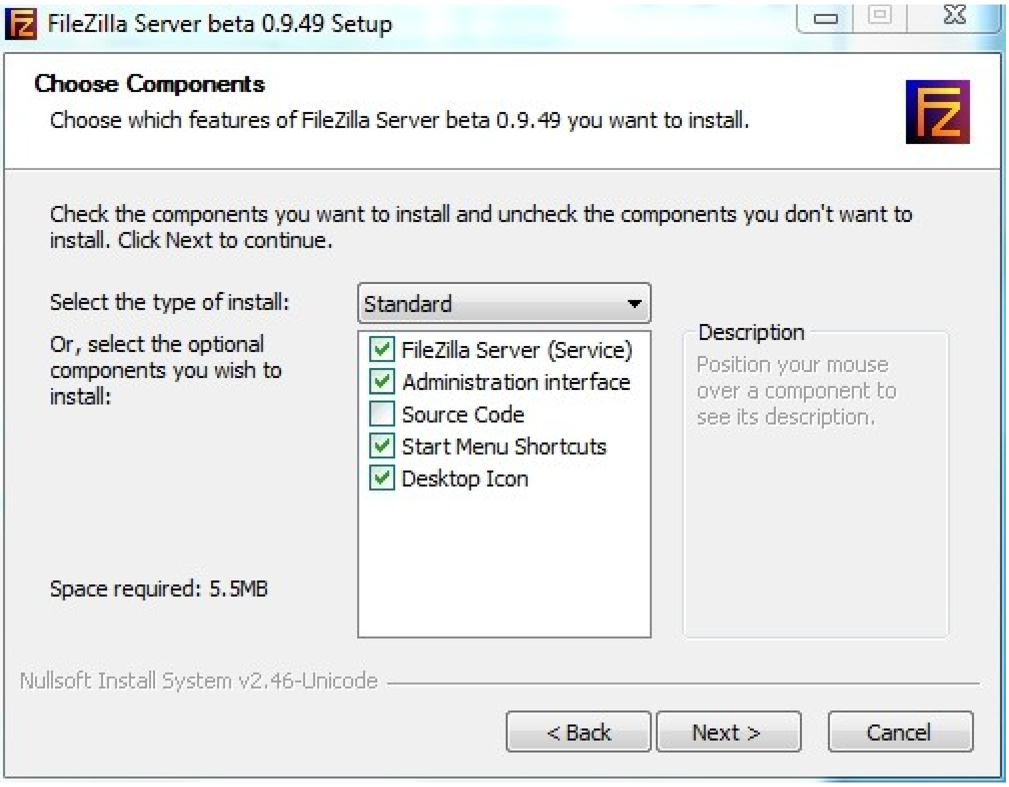
If you are using only the GSM-SMS communication, you don’t need to install FTP Server.



Filezilla FTP server:
Download free Filezilla FTP server from https://filezilla-project.org/

FileZilla_Server-0_9_49.exe

Install it to your PC using default settings.


FileZilla Server beta 0.9.49 Setup

Choose Components

Choose which features of FileZilla Server beta 0.9.49 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Standard

Or, select the optional components you wish to install:

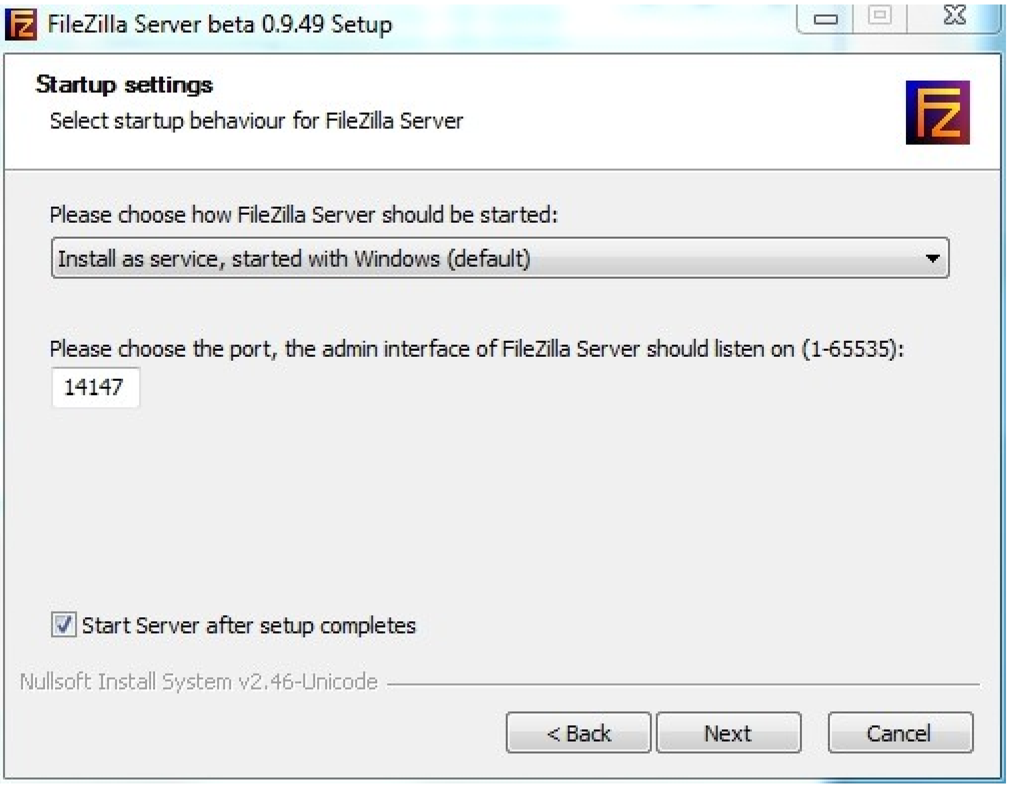
- ☑ FileZilla Server (Service)
- ☑ Administration interface
- □ Source Code
- ☑ Start Menu Shortcuts
- ☑ Desktop Icon

Description

Position your mouse over a component to see its description.

Space required: 5.5MB

Nullsoft Install System v2.46-Unicode



FileZilla Server beta 0.9.49 Setup

Startup settings

Select startup behaviour for FileZilla Server

Please choose how FileZilla Server should be started:

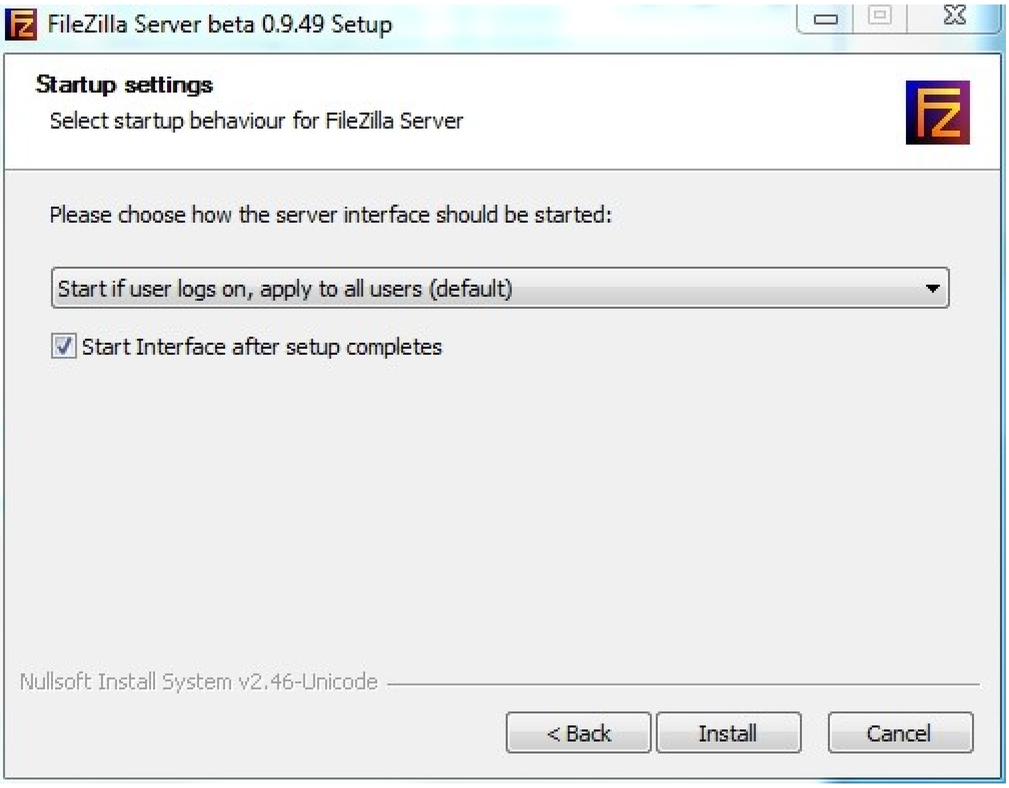
Install as service, started with Windows (default)

Please choose the port, the admin interface of FileZilla Server should listen on (1-65535):

14147

☑ Start Server after setup completes

Nullsoft Install System v2.46-Unicode



FileZilla Server beta 0.9.49 Setup

Startup settings

Select startup behaviour for FileZilla Server

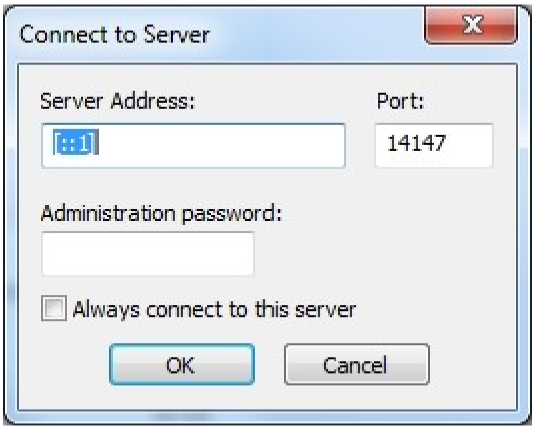
Please choose how the server interface should be started:

Start if user logs on, apply to all users (default)

☑ Start Interface after setup completes

Nullsoft Install System v2.46-Unicode

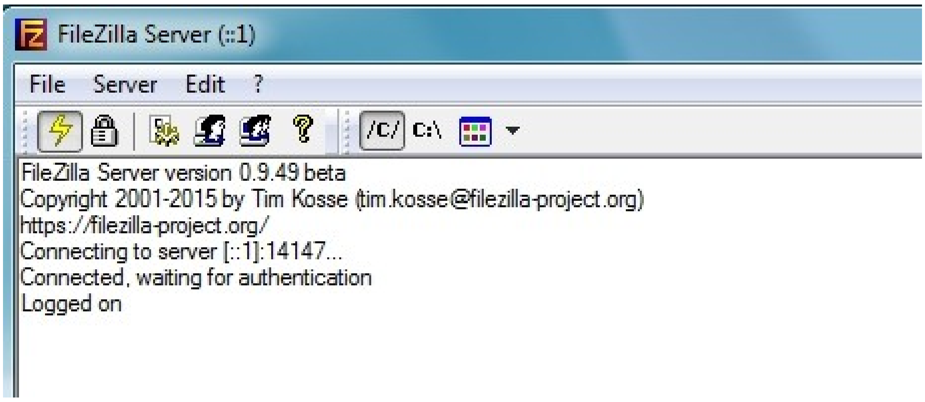
Installation is ready!



Server Address: [::1]
Port: 14147
Administration password:
☐ Always connect to this server

After that the FTP server starts automatically.

Click OK.

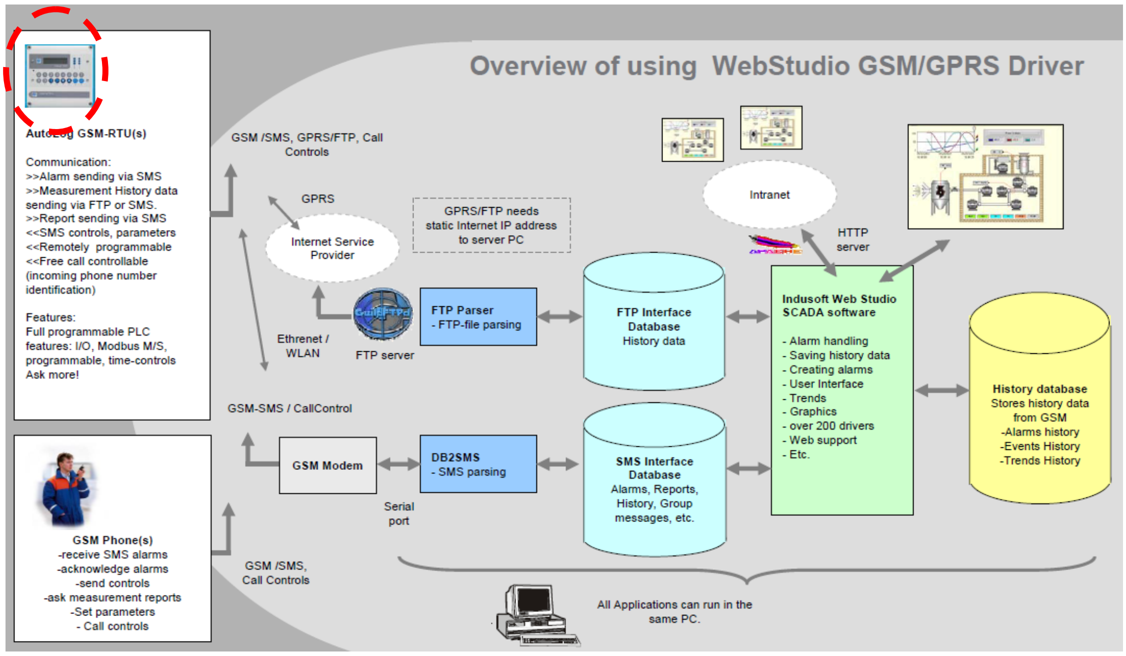


```
FileZilla Server version 0.9.49 beta
Copyright 2001-2015 by Tim Kosse (tim.kosse@filezilla-project.org)
https://filezilla-project.org/
Connecting to server [::1]:14147...
Connected, waiting for authentication
Logged on
```

The configuration of the FTP server is continued later.

4.6 Installing GSM Programmer

This is needed for both GPRS/FTP communication and GSM-SMS communication.



Overview of using WebStudio GSM/GPRS Driver

AutoLog GSM-RTU(s)

Communication:
>>Alarm sending via SMS
>>Measurement History data sending via FTP or SMS.
>>Report sending via SMS
<<SMS controls, parameters
<<Remotely programmable
<<Free call controllable (incoming phone number identification)

Features:
Full programmable PLC features: I/O, Modbus M/S, programmable, time-controls
Ask more!

GSM Phone(s)
-receive SMS alarms
-acknowledge alarms
-send controls
-ask measurement reports
-Set parameters
- Call controls

GSM /SMS, GPRS/FTP, Call Controls

GPRS

Internet Service Provider

GPRS/FTP needs static Internet IP address to server PC

Ethrenet / WLAN

FTP server

FTP Parser
- FTP-file parsing

GSM-SMS / CallControl

GSM Modem

Serial port

DB2SMS
- SMS parsing

GSM /SMS, Call Controls

FTP Interface Database
History data

SMS Interface Database
Alarms, Reports, History, Group messages, etc.

Intranet

HTTP server

Indusoft Web Studio SCADA software
- Alarm handling
- Saving history data
- Creating alarms
- User Interface
- Trends
- Graphics
- over 200 drivers
- Web support
- Etc.

History database
Stores history data from GSM
-Alarms history
-Events History
-Trends History

All Applications can run in the same PC.

4.6.1 Download latest version of GsmProgrammer

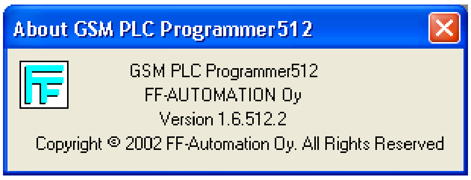
Latest version can be downloaded from the FF-Automation Web page's Distributor Area.

http://www.ff-automation.com/download/Distributor_signin.shtml

Please ask password from FF-Automation Oy!

E.g. to: antti.moijanen@ff-automation.com cc: info@ff-automation.com

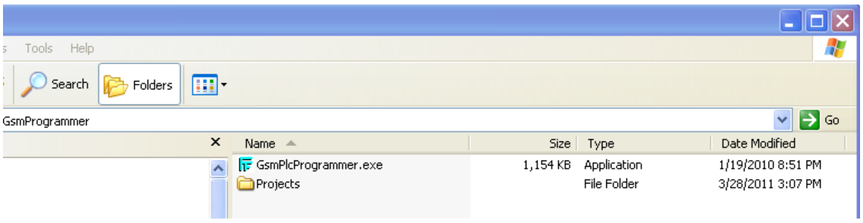
Latest version of GsmProgrammer while writing this manual is v. 1.6.512.2



About GSM PLC Programmer512

GSM PLC Programmer512
FF-AUTOMATION Oy
Version 1.6.512.2
Copyright © 2002 FF-Automation Oy. All Rights Reserved

4.6.2 Installing the software



Name	Size	Type	Date Modified
GsmPlcProgrammer.exe	1,154 KB	Application	1/19/2010 8:51 PM
Projects		File Folder	3/28/2011 3:07 PM

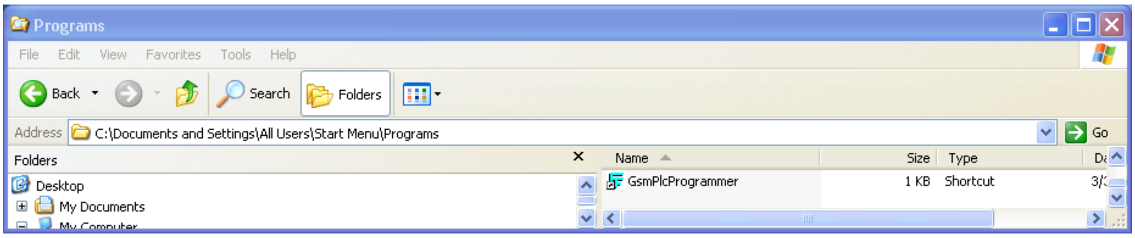
Create directory for GsmProgrammer (E.g. C:\GsmProgrammer).

Copy GsmProgrammer.exe into this directory.

Create new "Projects" folder under this directory.

4.6.3 Creating shortcut in Windows Start menu

Right click on “Programmer.exe” and select copy. (In Windows XP) Go to the folder C:\Documents and Settings\All Users\Start Menu\Programs and right click and select paste shortcut.

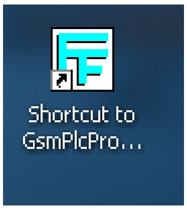


Address: C:\Documents and Settings\All Users\Start Menu\Programs

Name	Size	Type	Da
GsmPlcProgrammer	1 KB	Shortcut	3/

4.6.4 Creating shortcut in Windows desktop

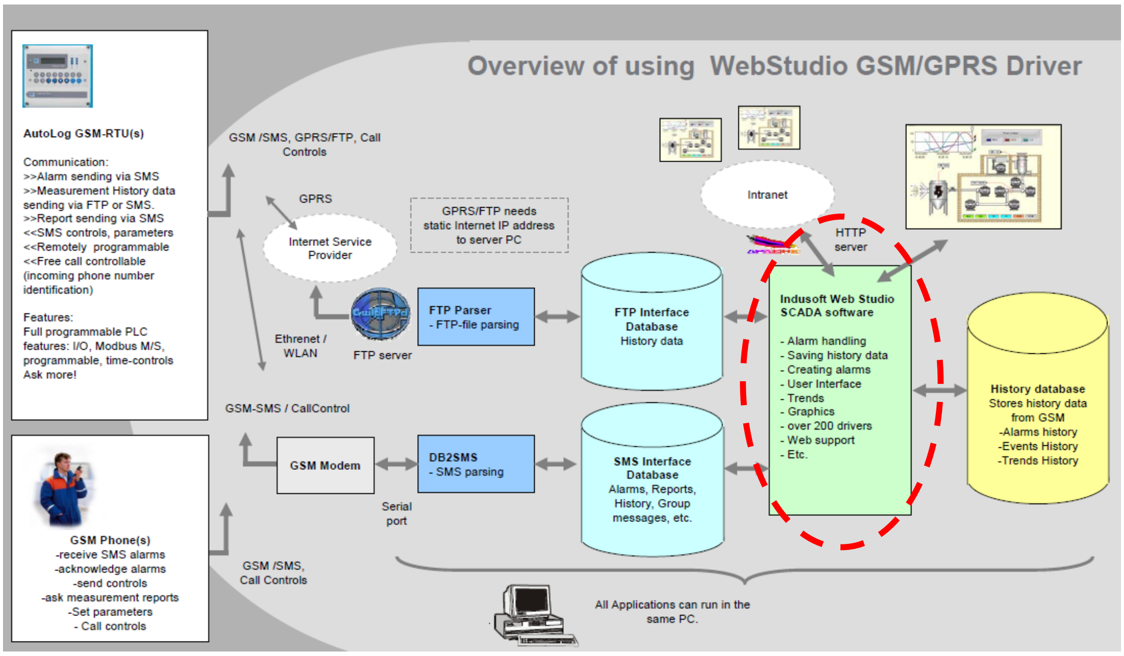
Right click on “Programmer.exe” and select copy. Go to desktop and right click and select paste shortcut.



Now you have installed GsmProgrammer!

4.7 Installing Indusoft Web Studio SCADA software

This is needed for both GPRS/FTP communication and GSM-SMS communication.



Download latest version of Indusoft Web Studio from Indusoft's web page.

www.indusoft.com

(You need to register to the page)

Or you can ask installation DVDs from FF-Automation.

Install the software according to IWS technical reference manual. (Log in to Windows OS with administrator user rights when you install the software).

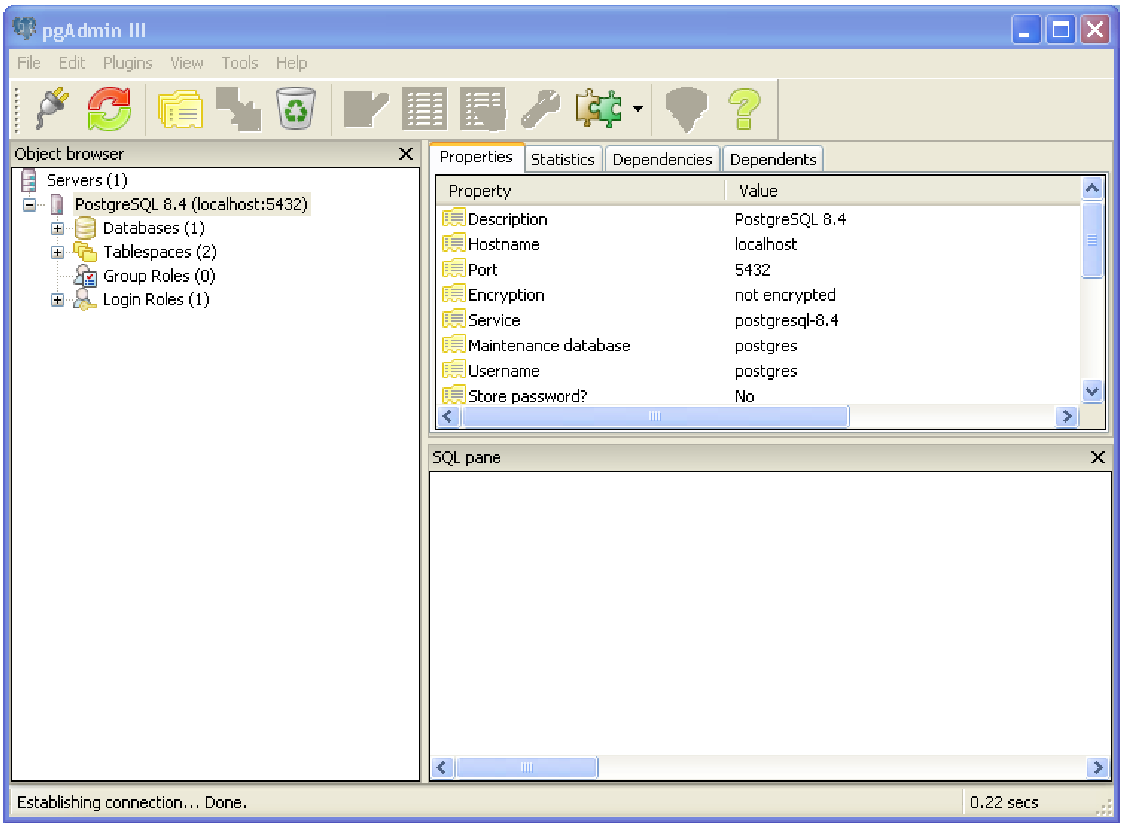
You can buy licenses to this software from FF-Automation.
www.ff-automation.com

5 Configuring the components

5.1 Configuring PostgreSQL database

This is needed for both GPRS/FTP communication and GSM-SMS communication.

1. You can configure the database by using a graphical (pgAdmin) or a console (psql) interface.
   a. Start pgAdmin III program, which can be used to configure database: Start -> Programs -> PostgreSQL -> pgAdmin III
   b. Start -> Programs -> PostgreSQL -> psql to ’postgres’



Servers (1)
- PostgreSQL 8.4 (localhost:5432)
  - Databases (1)
  - Tablespaces (2)
  - Group Roles (0)
  - Login Roles (1)

Property	Value
Description	PostgreSQL 8.4
Hostname	localhost
Port	5432
Encryption	not encrypted
Service	postgresql-8.4
Maintenance database	postgres
Username	postgres
Store password?	No

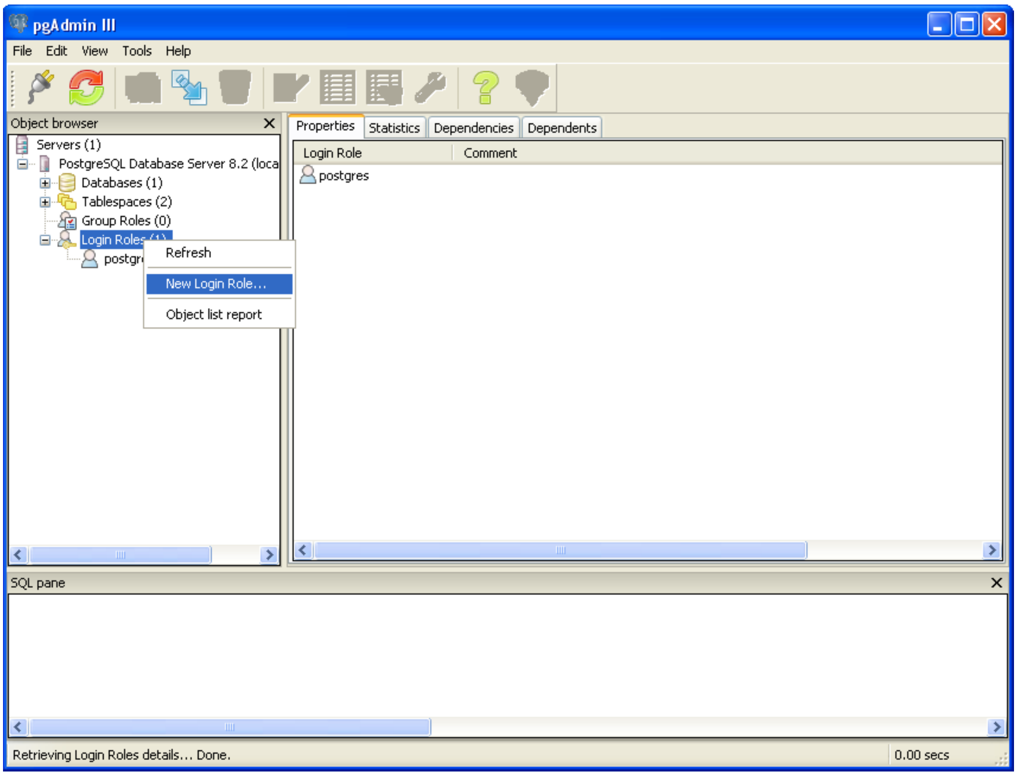
Establishing connection... Done. 0.22 secs

Note: If you want to change the language of pgAdmin’s to English, you can select: File -> Options -> General -> User Language -> English

2. Double click the serveron the left side. After this logon window opens. Write the superuser password to logon.

Psql guide: (This is just an optional alternative guide if you don’t like pgAdmin III. PgAdmin III is suggested for doing configurations.
Psql password is asked when you start the program.

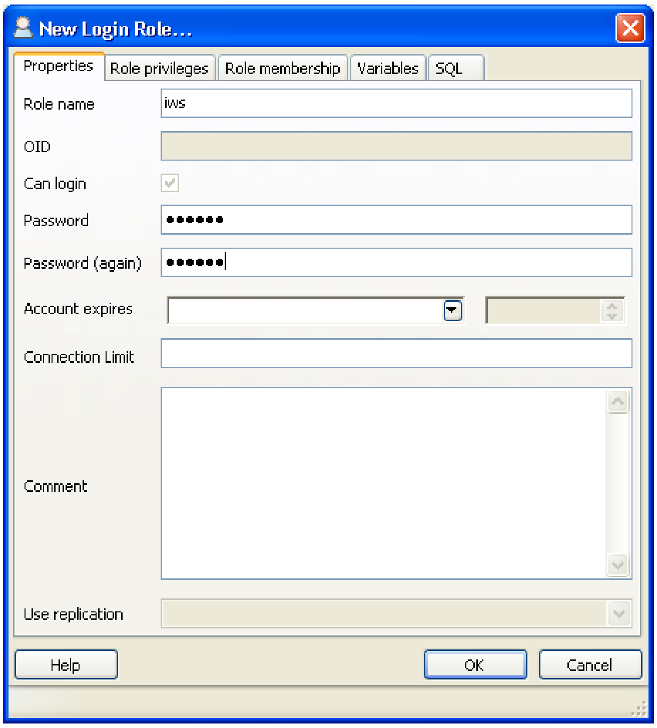
3. Create new role for databases. Select: Login Roles->New login role...



Login Role	Comment
postgres	

Retrieving Login Roles details... Done. 0.00 secs

4. Write to the New Login role window: Role Name: “iws” Password “your_own_password” (Remember the password!) and click *OK.*



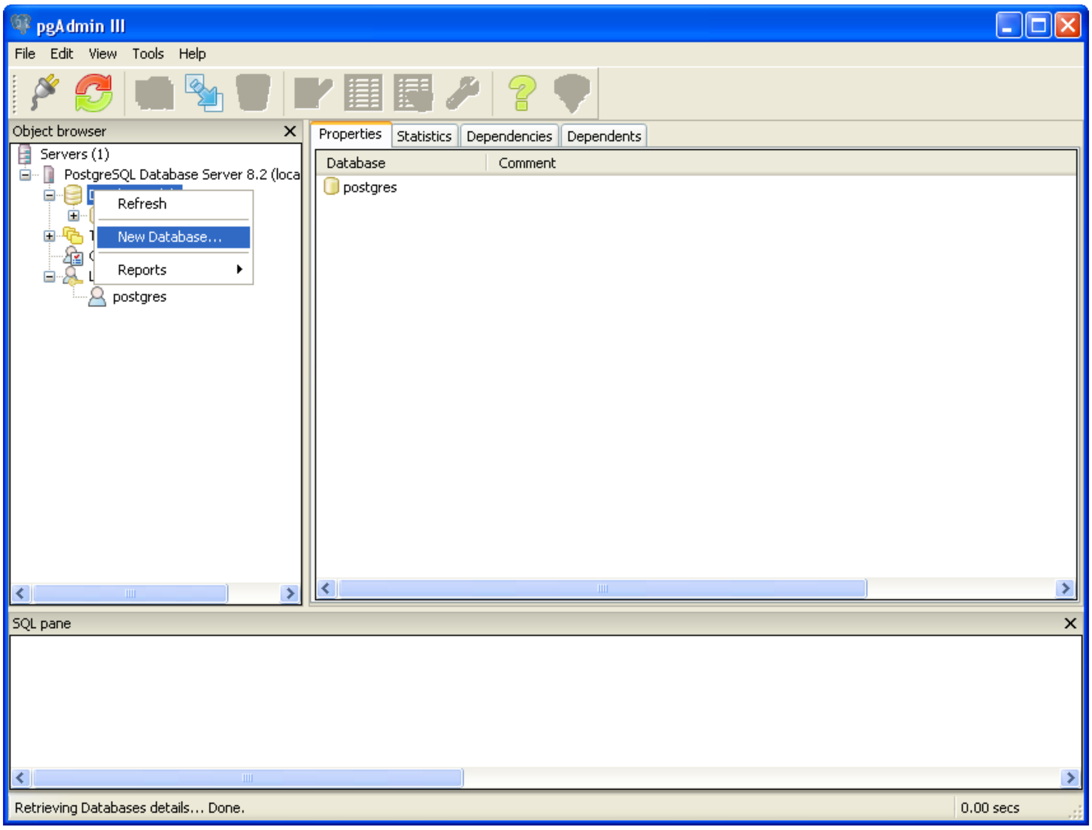
Role name	iws
OID	
Can login	☑
Password	••••••
Password (again)	••••••
Account expires	
Connection Limit	
Comment	
Use replication	

Psql guide:
With psql you can create new user using the following command:

```
CREATE ROLE iws LOGIN PASSWORD 'your_own_pw' NOINHERIT
VALID UNTIL 'infinity';
```

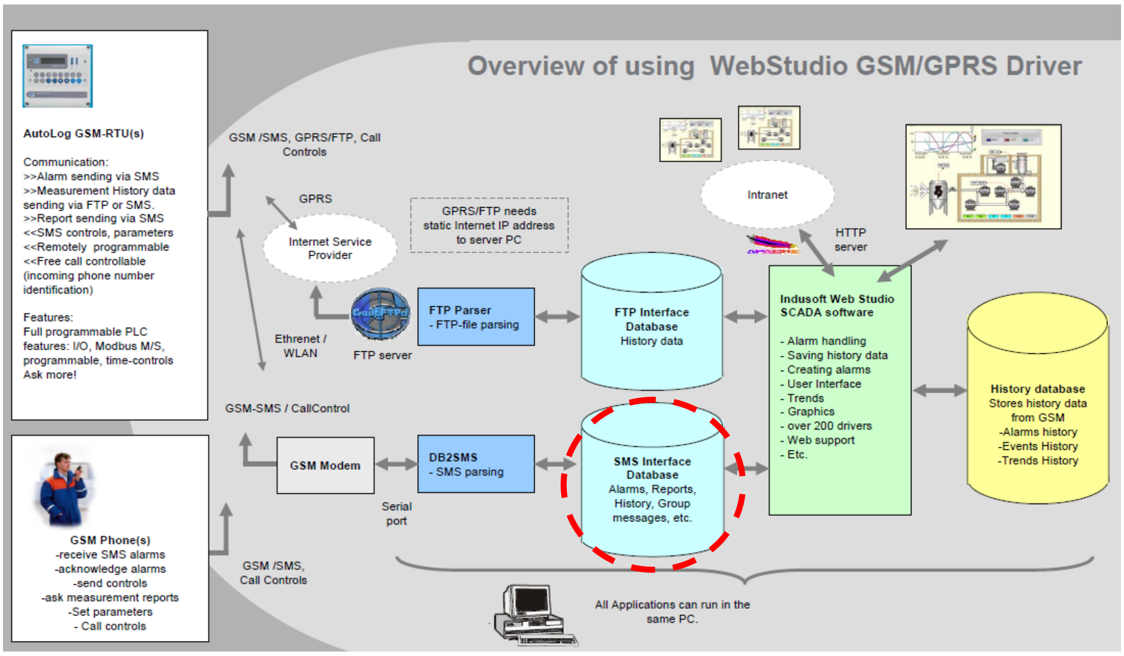
Note the semi colon (;) at the end of the line!

5. Creating a database. Right click on the Databases –text and select New database...



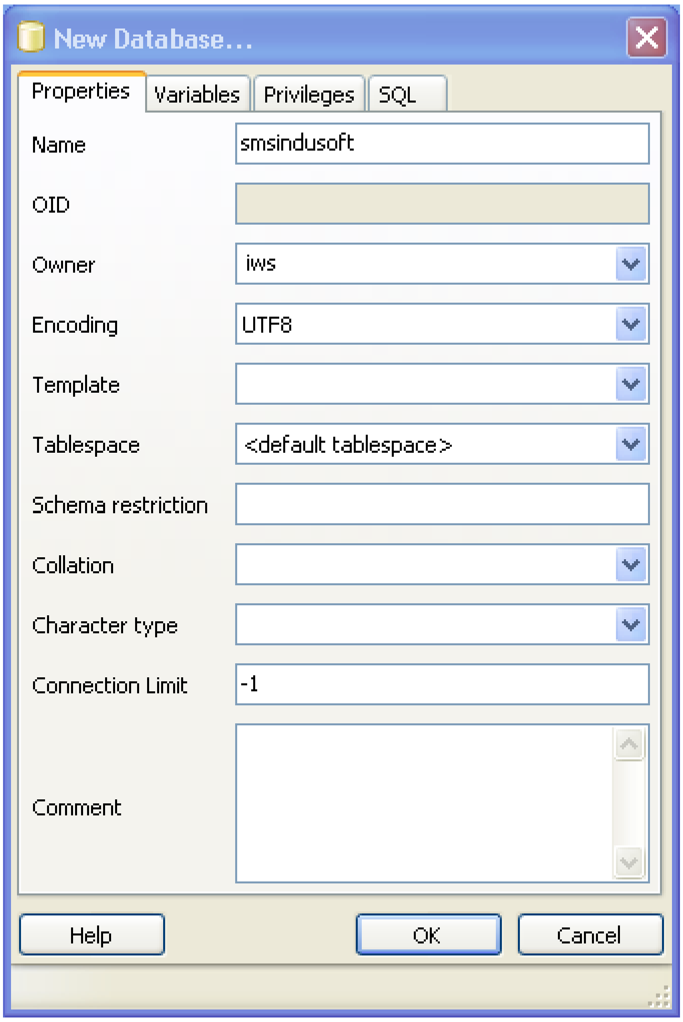
5.1.1 Creating database for DB2SMS driver (for GSM-SMS driver)

This is needed for GSM-SMS communication only if you are using GPRS/FTP communication you can skip this step.



1. *New Database...* –window opens. **Write Name: *smsindusoft* and Owner: *iws.***

- *Note: This database is needed only for DB2SMS driver*



Name: smsindusoft

OID:

Owner: iws

Encoding: UTF8

Template:

Tablespace: <default tablespace>

Schema restriction:

Collation:

Character type:

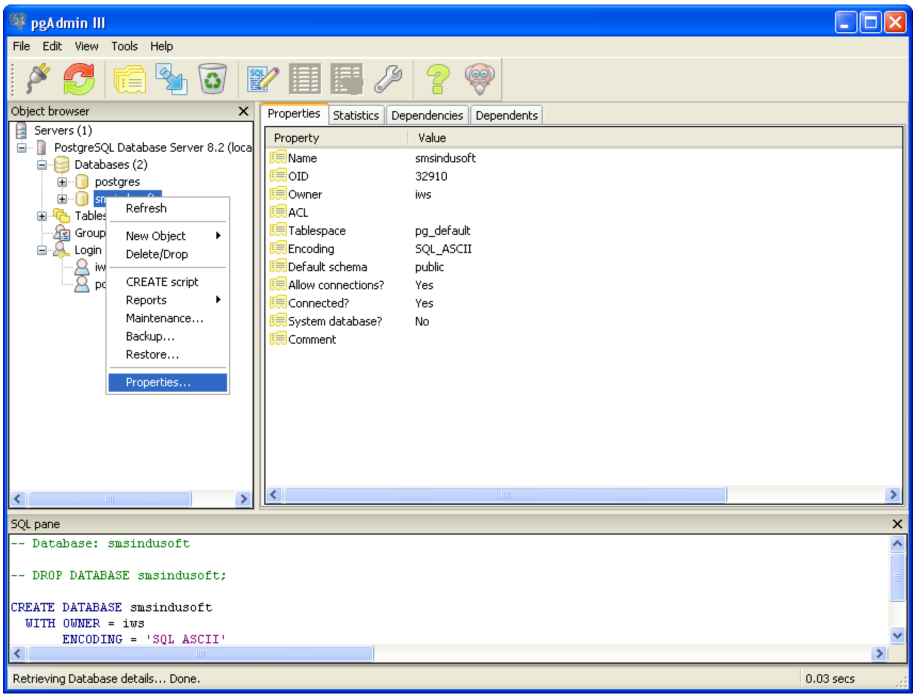
Connection Limit: -1

Comment:

Psql guide:
smsindusoft database can be created using the following command:

```
CREATE DATABASE smsindusoft WITH ENCODING='UTF8'
OWNER=iws;
```

2. Right click on the ”smsindusoft” database and select *Properties...*

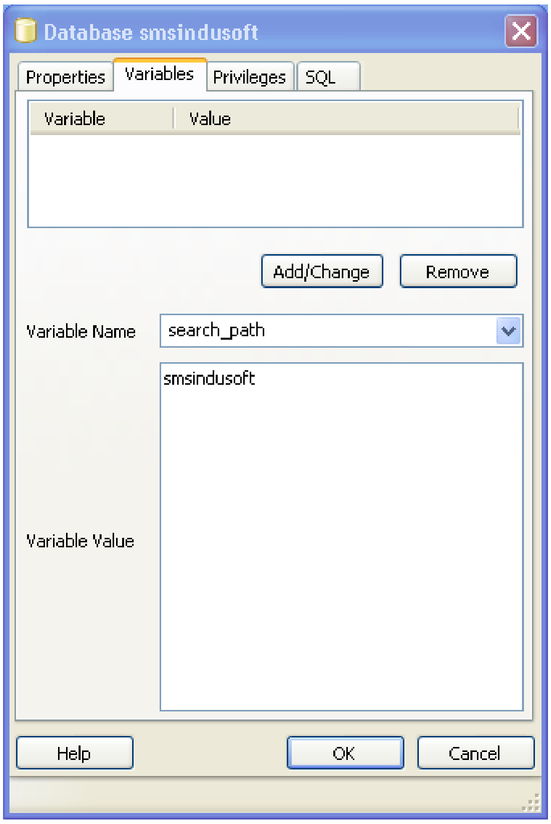


Property	Value
Name	smsindusoft
OID	32910
Owner	iws
ACL	
Tablespace	pg_default
Encoding	SQL_ASCII
Default schema	public
Allow connections?	Yes
Connected?	Yes
System database?	No
Comment	

```
-- Database: smsindusoft

-- DROP DATABASE smsindusoft;

CREATE DATABASE smsindusoft
  WITH OWNER = iws
       ENCODING = 'SQL_ASCII'
```

Retrieving Database details... Done. 0.03 secs

3. Select Variables –sheet and write *Variable name* ”search_path” and *Variable Value* ”smsindusoft”. Click Add/Change.



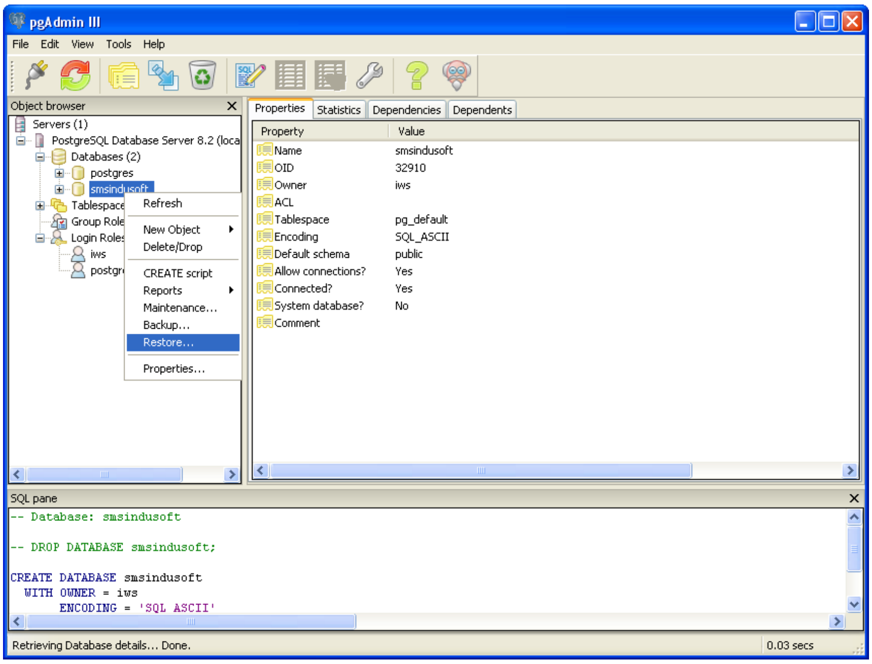
Variable Name: search_path

Variable Value: smsindusoft

Psql guide:
Setting search_path variable:

```
ALTER DATABASE smsindusoft SET search_path=smsindusoft;
```

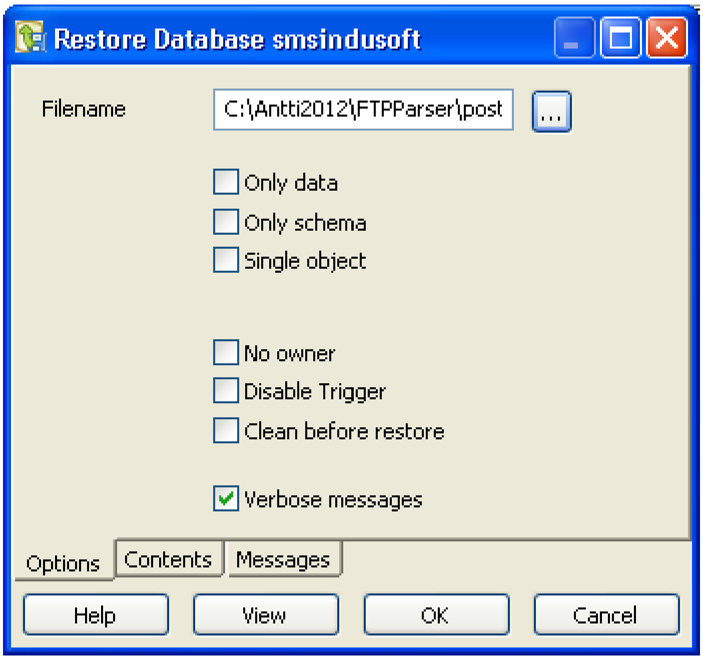
4. Right click smsindusoft –database and select Restore...



Property	Value
Name	smsindusoft
OID	32910
Owner	iws
ACL	
Tablespace	pg_default
Encoding	SQL_ASCII
Default schema	public
Allow connections?	Yes
Connected?	Yes
System database?	No
Comment	

```
-- Database: smsindusoft

-- DROP DATABASE smsindusoft;

CREATE DATABASE smsindusoft
  WITH OWNER = iws
       ENCODING = 'SQL_ASCII'
```

Retrieving Database details... Done. 0.03 secs



Filename C:\Antti2012\FTPParser\post

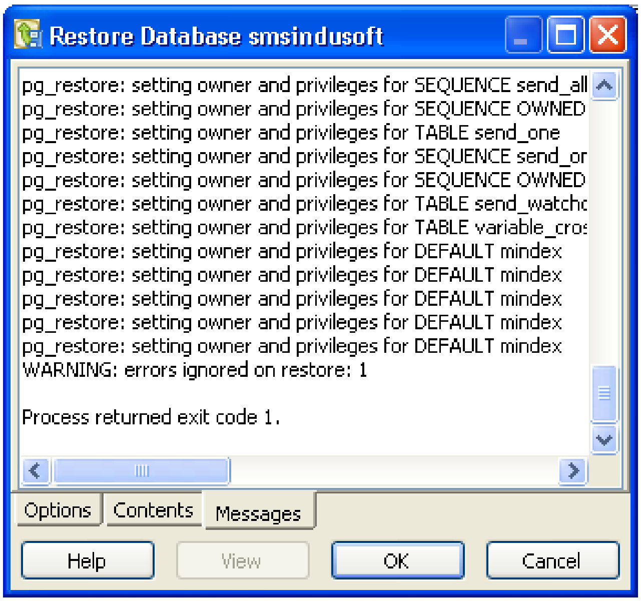
□ Only data
□ Only schema
□ Single object

□ No owner
□ Disable Trigger
□ Clean before restore

☑ Verbose messages

5. From the opening window browse and select smsindusoft_2013.backup –file. This file includes all the tables and default values needed for DB2SMS so you don't need to re-write those. smsindusoft_2013.backup –file is sent with the DB2SMS driver -package. When you are ready click OK.

6. Database restore –window after successful restore



```
pg_restore: setting owner and privileges for SEQUENCE send_all
pg_restore: setting owner and privileges for SEQUENCE OWNED
pg_restore: setting owner and privileges for TABLE send_one
pg_restore: setting owner and privileges for SEQUENCE send_or
pg_restore: setting owner and privileges for SEQUENCE OWNED
pg_restore: setting owner and privileges for TABLE send_watchc
pg_restore: setting owner and privileges for TABLE variable_cros
pg_restore: setting owner and privileges for DEFAULT mindex
pg_restore: setting owner and privileges for DEFAULT mindex
pg_restore: setting owner and privileges for DEFAULT mindex
pg_restore: setting owner and privileges for DEFAULT mindex
pg_restore: setting owner and privileges for DEFAULT mindex
WARNING: errors ignored on restore: 1

Process returned exit code 1.
```

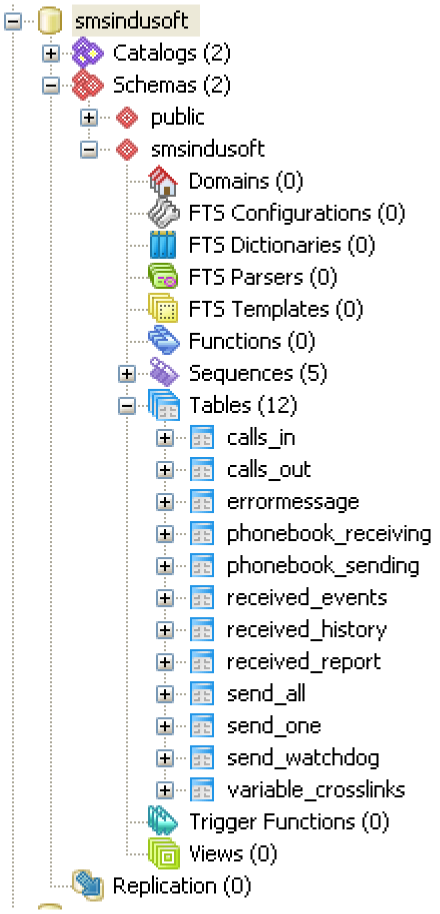
7. Configuring PostgreSQL is finished.

Psql guide:
Restoring the database cannot be done in psql –command line tool, but it can be easily made in command prompt. (press Ctrl+C in psql). Note that you need to be in the same directory as the PostgreSQL management tools. Restore command:

```
 pg_restore.exe -i -h localhost -p 5432 -U postgres -d smsindusoft -v "C:\
DB2SMS\1.1\Databases\ smsindusoft_2013.backup"
```

The program asks for the password you used while installing PostgreSQL. After writing the password, press enter. You can ignore errors happened in restoring process.

After restore - Check that you have 12 tables created:



- smsindusoft
  - Catalogs (2)
  - Schemas (2)
    - public
    - smsindusoft
      - Domains (0)
      - FTS Configurations (0)
      - FTS Dictionaries (0)
      - FTS Parsers (0)
      - FTS Templates (0)
      - Functions (0)
      - Sequences (5)
      - Tables (12)
        - calls_in
        - calls_out
        - errormessage
        - phonebook_receiving
        - phonebook_sending
        - received_events
        - received_history
        - received_report
        - send_all
        - send_one
        - send_watchdog
        - variable_crosslinks
      - Trigger Functions (0)
      - Views (0)
  - Replication (0)

if the calls_in and calls_out tables are not created, you can open psql (shell) and open database smsindusoft and create the following tables:

```
CREATE TABLE calls_in (
 flag numeric,
 name text,
 phone text
) WITHOUT OIDS;
ALTER TABLE calls_in OWNER TO iws;

CREATE TABLE "calls_out" (
 flag numeric,
 name text,
 phone text
) WITHOUT OIDS;
ALTER TABLE calls_out OWNER TO iws;
```

5.1.2 Creating database for DB2SMS driver (for GSM-SMS driver)

-Note smshistory database is only needed for DB2SMS driver.

The history database is used by Indusoft Web Studio for storing handled history data, if you are using DB2SMS only for SMS alarm forwarding then you don’t need this database.

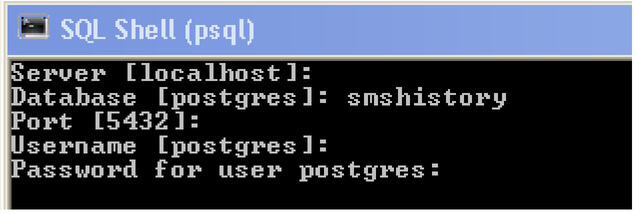
Creating the history database is done in the similar way as you did with smsindusoft-database. You can use the phases 1 to 3 of the former instructions simple by replacing smsindusoft with smshistory (search_path=smshistory)..

(steps 4-> for restoring the database are not needed unless you have demo database which you want to restore!)

The tables for history database has to be created manually. The easiest way to do this is doing it by using psql -tool

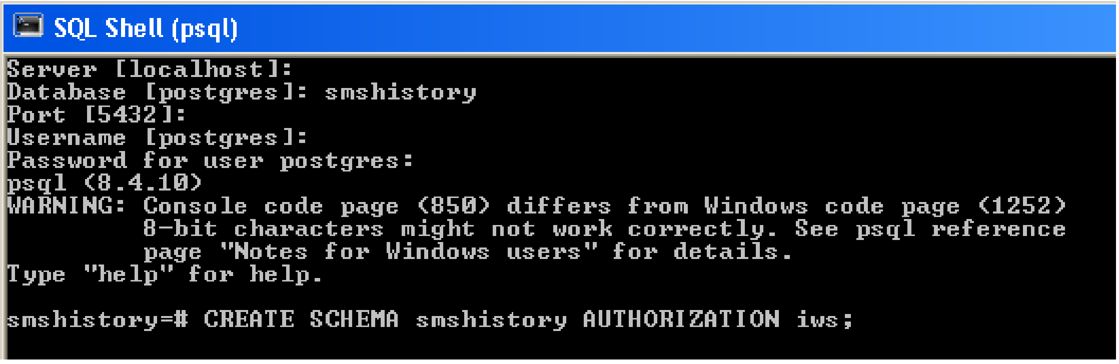
1. Open psql-tool

Start->All Programs->Postgre SQL->SQL shell (psql)



```
Server [localhost]:
Database [postgres]: smshistory
Port [5432]:
Username [postgres]:
Password for user postgres:
```

Click Enter to select Server [localhost]
Change database to smshistory
Click Enter to select Port [5432]
Click enter to select Username [postgres]
Write the password for user postgres. (it was defined during the installation process).



```
Server [localhost]:
Database [postgres]: smshistory
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (8.4.10)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

smshistory=# CREATE SCHEMA smshistory AUTHORIZATION iws;
```

2. Creating a new schema: CREATE SCHEMA smshistory AUTHORIZATION iws;

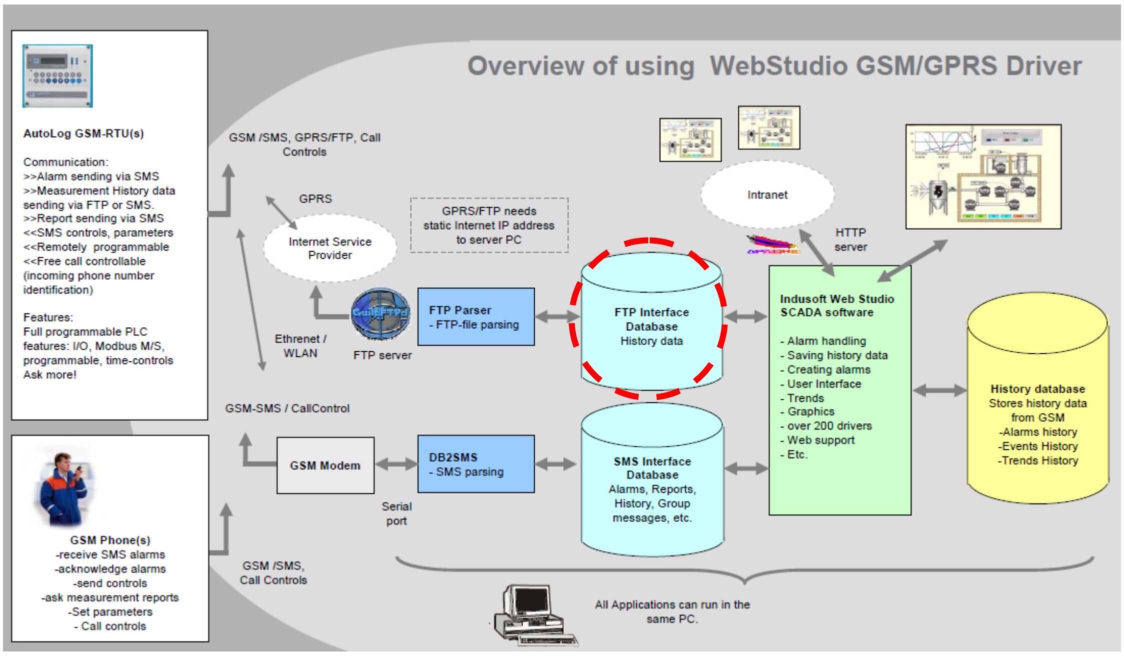
3. Creating a new table:

Example bellow:

```
CREATE TABLE trend_1
(
  time_stamp timestamp without time zone NOT NULL,
  time_stamp_ms integer DEFAULT 0,
  p1_run_time_24h integer,
  p1_start_count_24h integer,
  p2_run_time_24h integer,
  p2_start_count_24h integer,
  wm99 integer,
  ai0 real,
  ai1 real,
  ai2 real,
  ai3 real,
  isalarm integer DEFAULT 0,
  CONSTRAINT trend_1_pkey PRIMARY KEY (time_stamp)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE trend_1 OWNER TO iws;
```

5.1.3 Creating FTP interface database for FTP_Parser (GSM-GPRS/FTP file parsing)

This is needed for GPRS/FTP communication only if you are using GSM-SMS communication you can skip this step.



Overview of using WebStudio GSM/GPRS Driver

AutoLog GSM-RTU(s)

Communication:
>>Alarm sending via SMS
>>Measurement History data sending via FTP or SMS.
>>Report sending via SMS
<<SMS controls, parameters
<<Remotely programmable
<<Free call controllable (incoming phone number identification)

Features:
Full programmable PLC features: I/O, Modbus M/S, programmable, time-controls
Ask more!

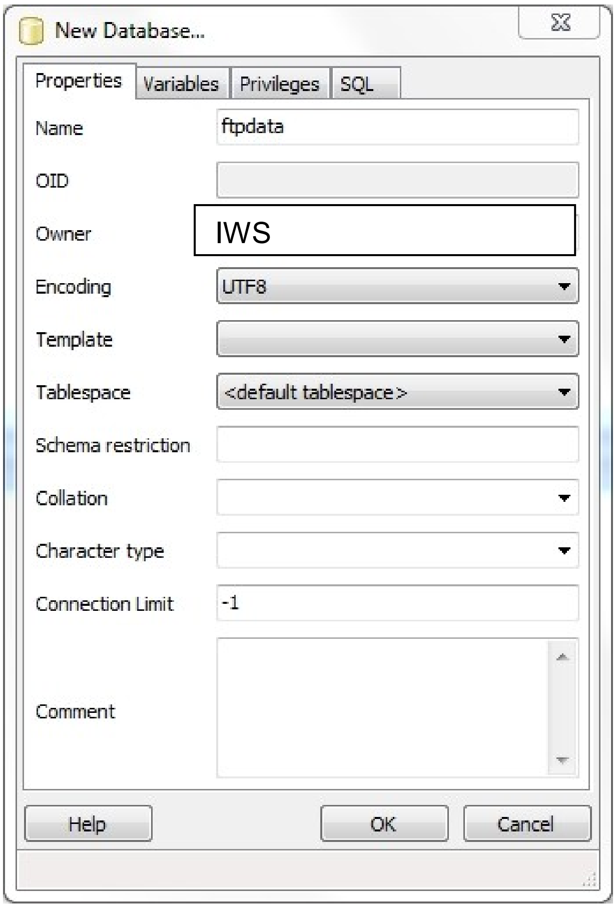
GSM Phone(s)
-receive SMS alarms
-acknowledge alarms
-send controls
-ask measurement reports
-Set parameters
- Call controls

Indusoft Web Studio SCADA software
- Alarm handling
- Saving history data
- Creating alarms
- User Interface
- Trends
- Graphics
- over 200 drivers
- Web support
- Etc.

History database
Stores history data from GSM
-Alarms history
-Events History
-Trends History

1. *New Database...* –window opens. Write Name: ***ftpdata* and Owner: *iws.***

- *Note: This database is needed only for FTP_Parser*

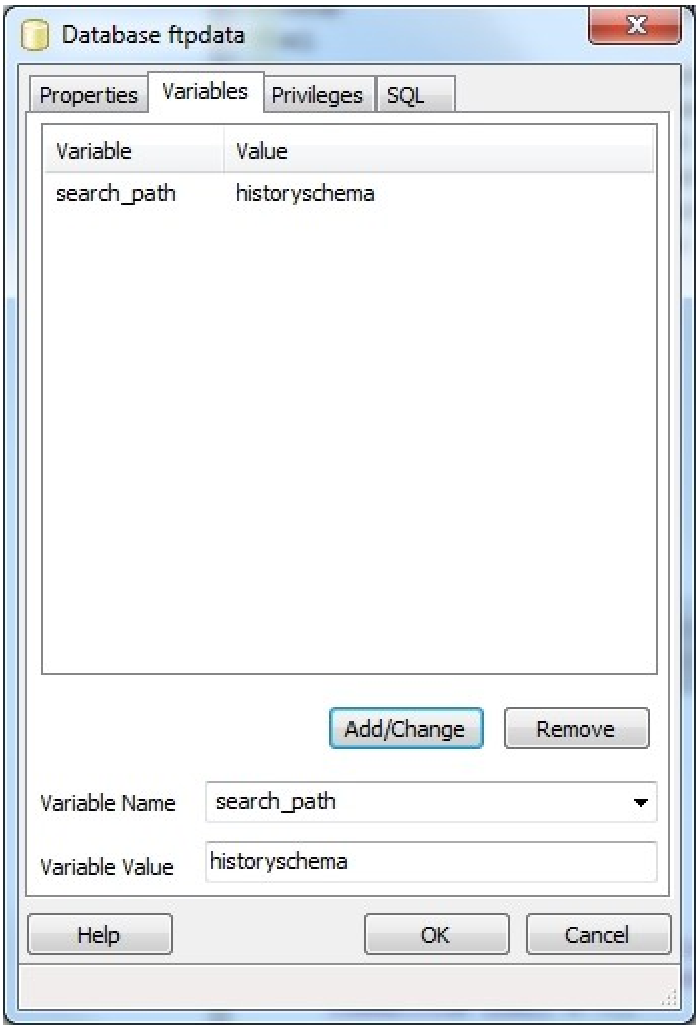


New Database...

Name	ftpdata
OID	
Owner	IWS
Encoding	UTF8
Template	
Tablespace	<default tablespace>
Schema restriction	
Collation	
Character type	
Connection Limit	-1
Comment	

2. Right click on the ”ftpdata” database and select *Properties...*

3. Select Variables –sheet and write *Variable name* ”search_path” and *Variable Value* ” historyschema”. Click Add/Change. Click OK.



Variable	Value
search_path	historyschema

Variable Name search_path

Variable Value historyschema

-Note! ftpdata database is only needed for FTP_file_parser driver.

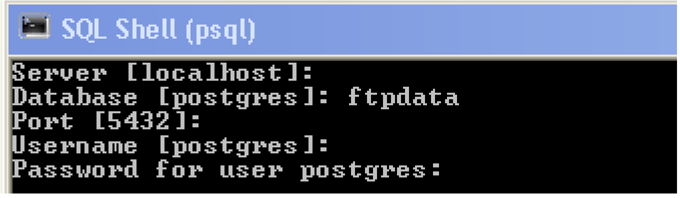
5.1.4 Creating tables for ftpdata –database interface

Restoring the ftpdata -database tables are not needed unless you have some already made template database which you want to reuse!) The ftpdata –database table columns depend on what variables you want to read from the the “GSM-PLC application”. Next example is showing how to create table to the FTP_Demo –application.

The tables of ftpdata database are normally created manually. The easiest way to do this is doing it by using psql -tool

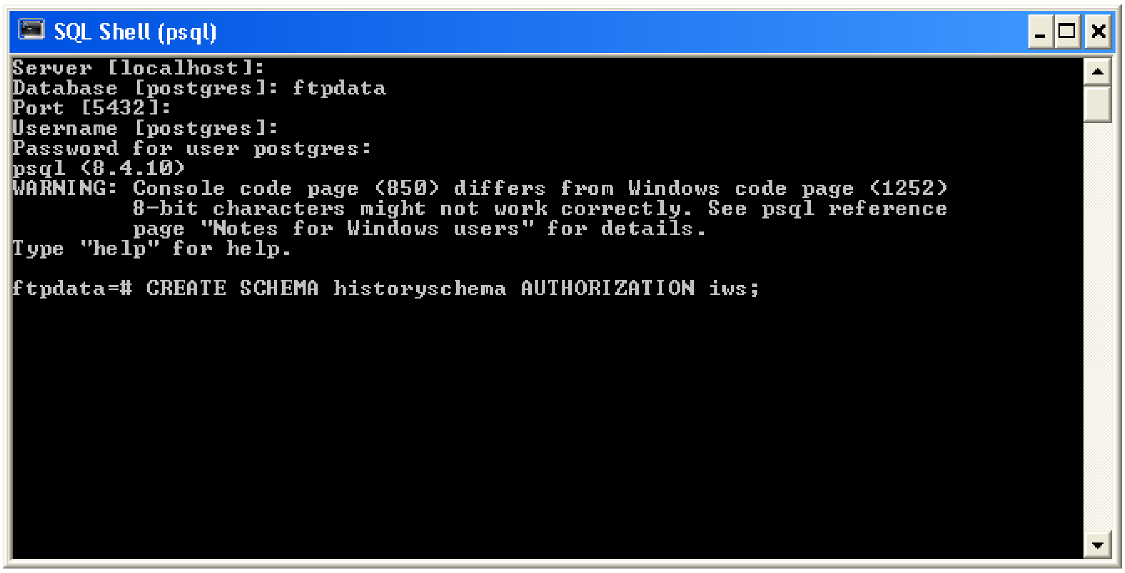
Open psql-tool

Start->All Programs->Postgre SQL->SQL shell (psql)



```
Server [localhost]:
Database [postgres]: ftpdata
Port [5432]:
Username [postgres]:
Password for user postgres:
```

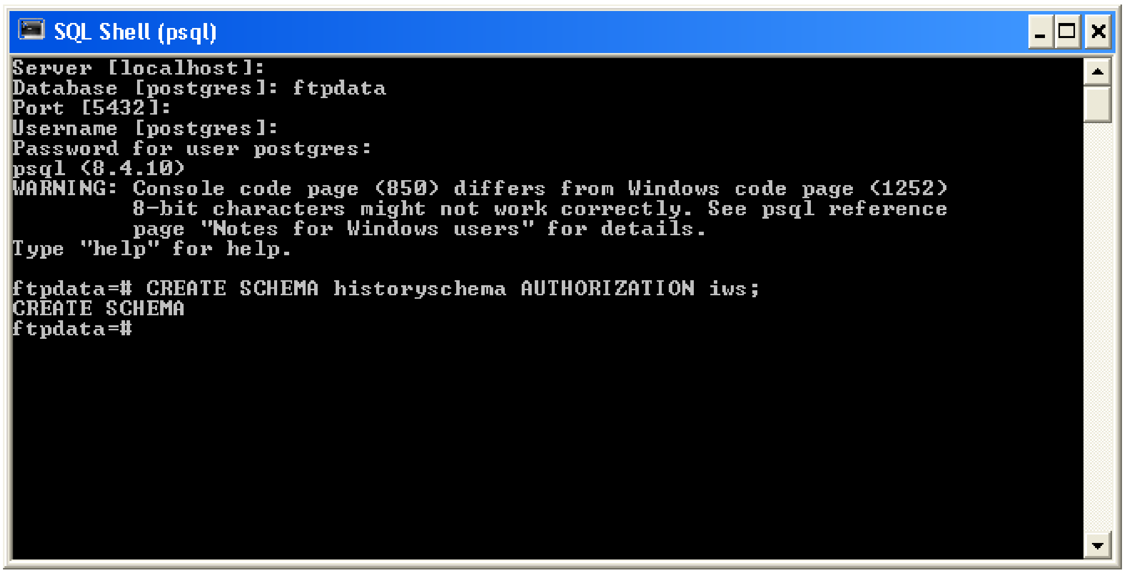
Click Enter to select Server [localhost]
Change database to ftpdata
Click Enter to select Port [5432]
Click enter to select Username [postgres]
Write the password for user postgres. (it was defined during the installation process).



```
Server [localhost]:
Database [postgres]: ftpdata
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (8.4.10)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

ftpdata=# CREATE SCHEMA historyschema AUTHORIZATION iws;
```

Write the following line to psql:

CREATE SCHEMA historyschema AUTHORIZATION iws;



```
Server [localhost]:
Database [postgres]: ftpdata
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (8.4.10)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

ftpdata=# CREATE SCHEMA historyschema AUTHORIZATION iws;
CREATE SCHEMA
ftpdata=#
```

Click Enter.

CONFIGURATIONS

Now you can create the table. The structure of the table depends on your PLC variable configuration. Replace *wm20 integer, wm21 integer, …* by your variable configuration and trend_*plcID* by your PLC ID. For testing the FTP DEMO application, you should not change the default values.

So for example:

```
CREATE TABLE trend_ffa00001
(
  time_stamp timestamp without time zone NOT NULL,
  time_stamp_ms integer,
  wm20 integer,
  wm21 integer,
  wm22 integer,
  wm23 integer,
  wm99 integer,
  wm100 integer,
  ro98 integer,
  CONSTRAINT trend_ffa00001_pkey PRIMARY KEY (time_stamp)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE trend_ffa00001 OWNER TO iws;
```

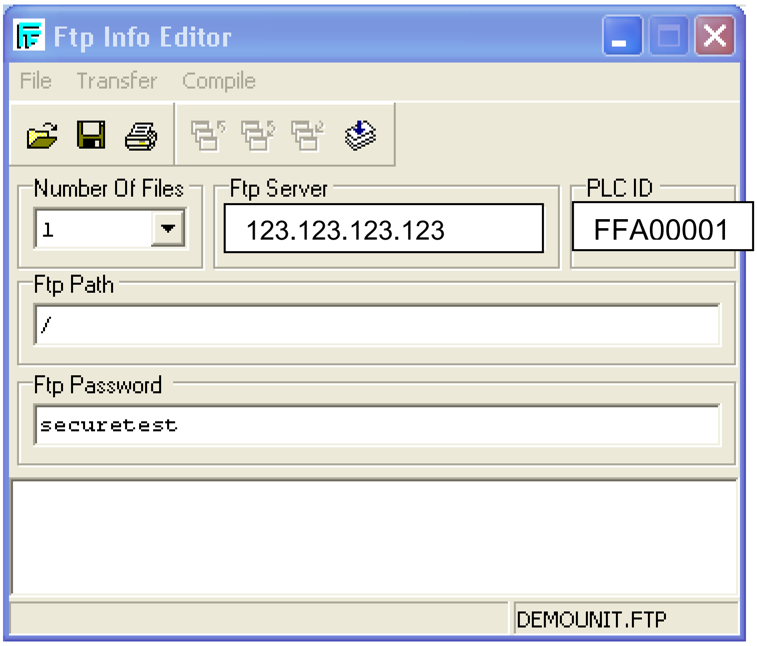
Table name should be

”trend_” + [PLC_ID].

Look bellow!

Note! In FTP_Parser interface table you cannot use row names like WM21_name, you can use only standard names like WM21. Look FileParser description (appendix) for more information!
Note2! The above table is used in GSM-PLC Demo project.

(PLC_ID can be set using GSMProgrammer. View-> FTP info)



Number Of Files: 1
Ftp Server: 123.123.123.123
PLC ID: FFA00001
Ftp Path: /
Ftp Password: securetest
DEMOUNIT.FTP

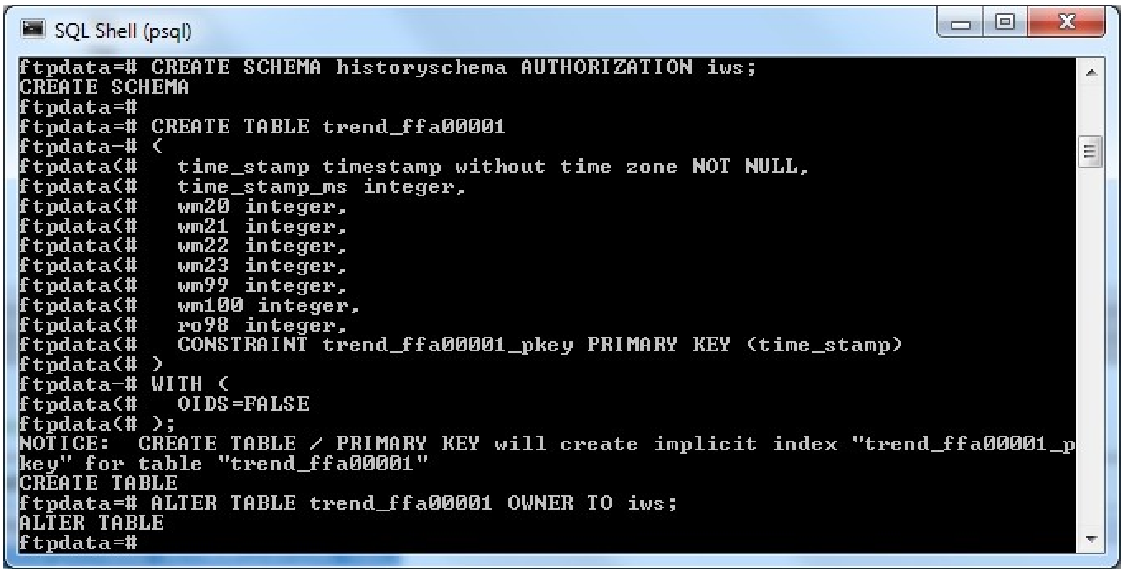
PLC_ID should have **8 HEX characters!**

(HEX characters are 0123456789ABCDEF)

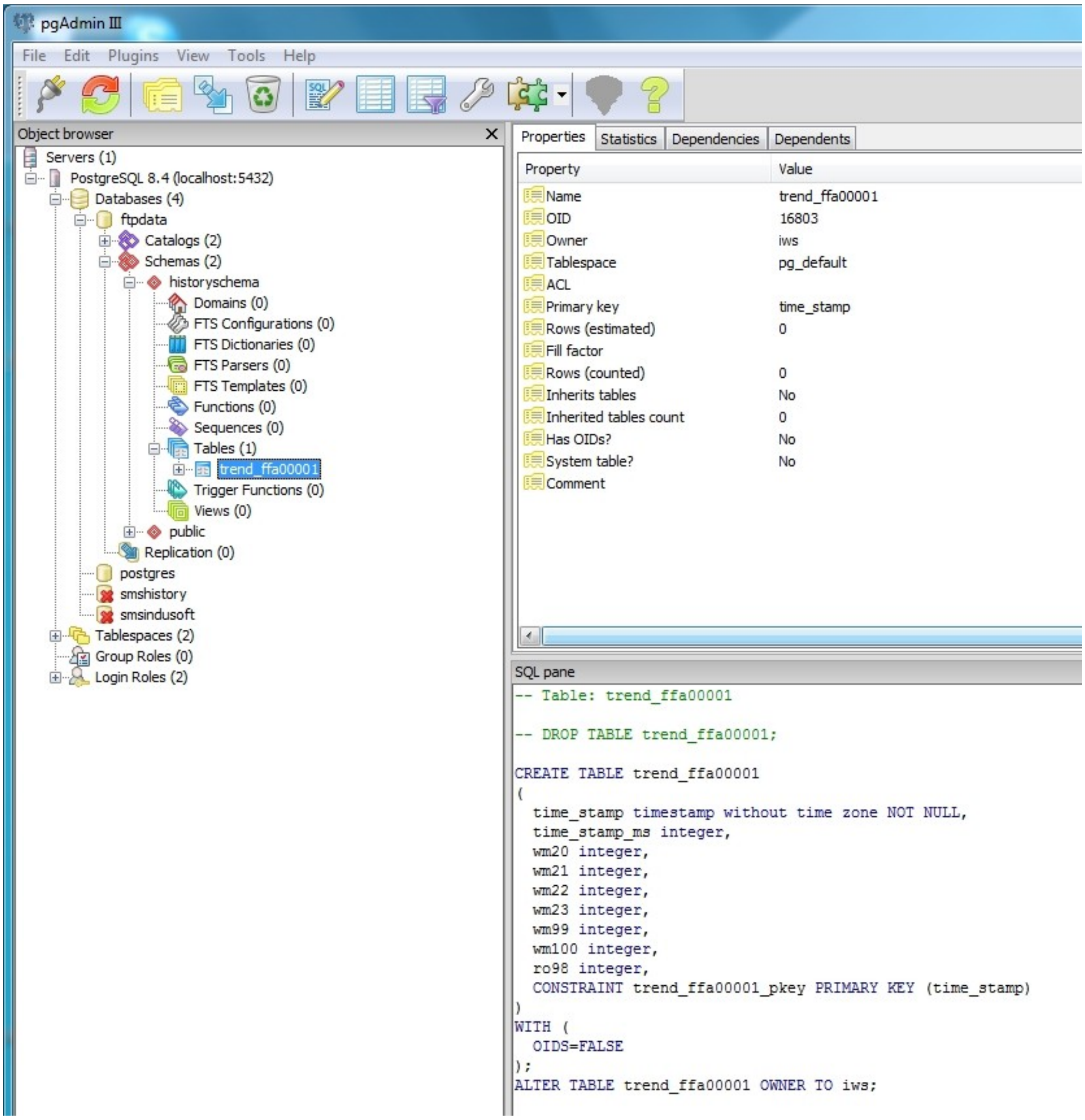
(After changing the PLC ID, you need to compile the project and transfer it to GSM-PLC and start the PLC, Look GSM-PLC User Manual how to do that!
Every GSM-PLC you have, should have unique PLC_ID, you can use numbers 0-9 and letters A-F.

Copy-paste the “CREATE TABLE” –command from this manual to psql. Click Enter.

You can copy-paste this command also from the .txt file form the *Manual _accessories_files* (sub directory)



```
ftpdata=# CREATE SCHEMA historyschema AUTHORIZATION iws;
CREATE SCHEMA
ftpdata=#
ftpdata=# CREATE TABLE trend_ffa00001
ftpdata-# (
ftpdata(#   time_stamp timestamp without time zone NOT NULL,
ftpdata(#   time_stamp_ms integer,
ftpdata(#   wm20 integer,
ftpdata(#   wm21 integer,
ftpdata(#   wm22 integer,
ftpdata(#   wm23 integer,
ftpdata(#   wm99 integer,
ftpdata(#   wm100 integer,
ftpdata(#   ro98 integer,
ftpdata(#   CONSTRAINT trend_ffa00001_pkey PRIMARY KEY (time_stamp)
ftpdata(# )
ftpdata-# WITH (
ftpdata(#   OIDS=FALSE
ftpdata(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "trend_ffa00001_p
key" for table "trend_ffa00001"
CREATE TABLE
ftpdata=# ALTER TABLE trend_ffa00001 OWNER TO iws;
ALTER TABLE
ftpdata=#
```



Property	Value
Name	trend_ffa00001
OID	16803
Owner	iws
Tablespace	pg_default
ACL	
Primary key	time_stamp
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Has OIDs?	No
System table?	No
Comment	

```
-- Table: trend_ffa00001

-- DROP TABLE trend_ffa00001;

CREATE TABLE trend_ffa00001
(
  time_stamp timestamp without time zone NOT NULL,
  time_stamp_ms integer,
  wm20 integer,
  wm21 integer,
  wm22 integer,
  wm23 integer,
  wm99 integer,
  wm100 integer,
  ro98 integer,
  CONSTRAINT trend_ffa00001_pkey PRIMARY KEY (time_stamp)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE trend_ffa00001 OWNER TO iws;
```

Open pgAdmin and you can view just created schema and table. If you cannot see the schema or tables, click refresh –button.

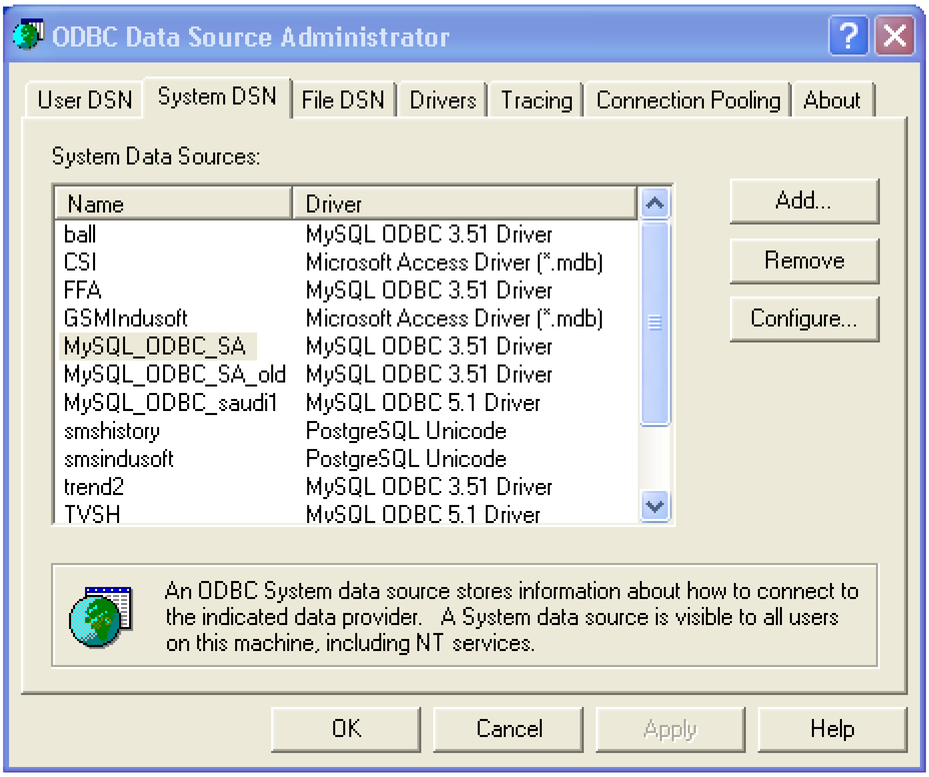
5.2 Configuring ODBC database Data Source Names (DNS)

This is need for both GSM-SMS and GPRS/FTP.

Create ODBC-connections for PostgreSQL, -databases:

Start->ControlPanel->Administrative Tools->Data sources (ODBC)

(If you have 64-bit Windows 7, you need to use the following program to configure 32-bit ODBC: c:\windows\syswow64\odbcad32.exe)

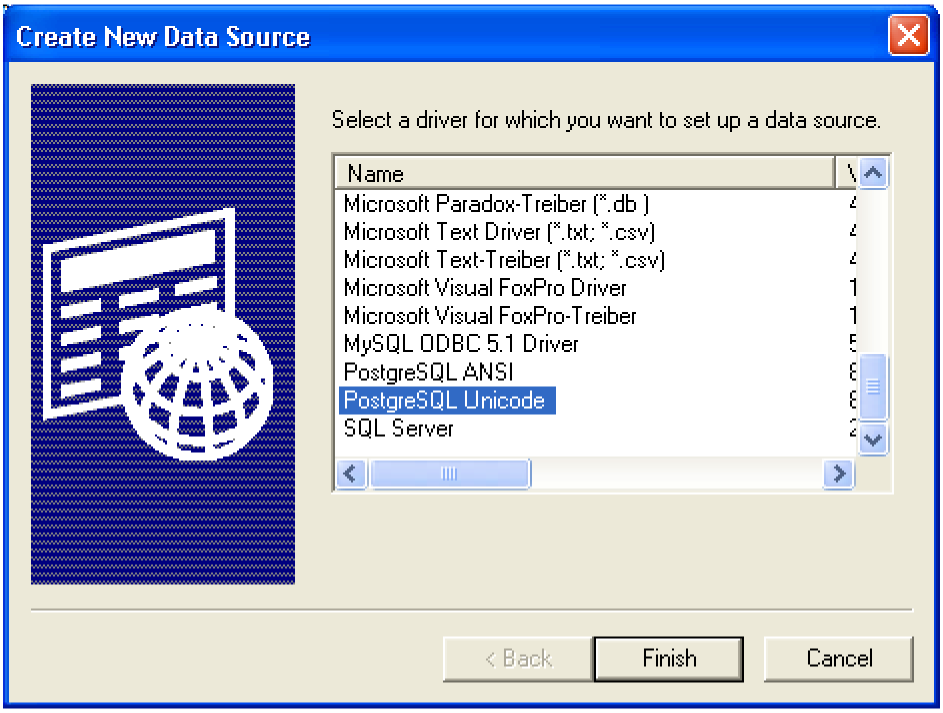


System Data Sources:

Name	Driver
ball	MySQL ODBC 3.51 Driver
CSI	Microsoft Access Driver (*.mdb)
FFA	MySQL ODBC 3.51 Driver
GSMIndusoft	Microsoft Access Driver (*.mdb)
MySQL_ODBC_SA	MySQL ODBC 3.51 Driver
MySQL_ODBC_SA_old	MySQL ODBC 3.51 Driver
MySQL_ODBC_saudi1	MySQL ODBC 5.1 Driver
smshistory	PostgreSQL Unicode
smsindusoft	PostgreSQL Unicode
trend2	MySQL ODBC 3.51 Driver
TVSH	MySQL ODBC 5.1 Driver

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Open System DNS -tab
click Add



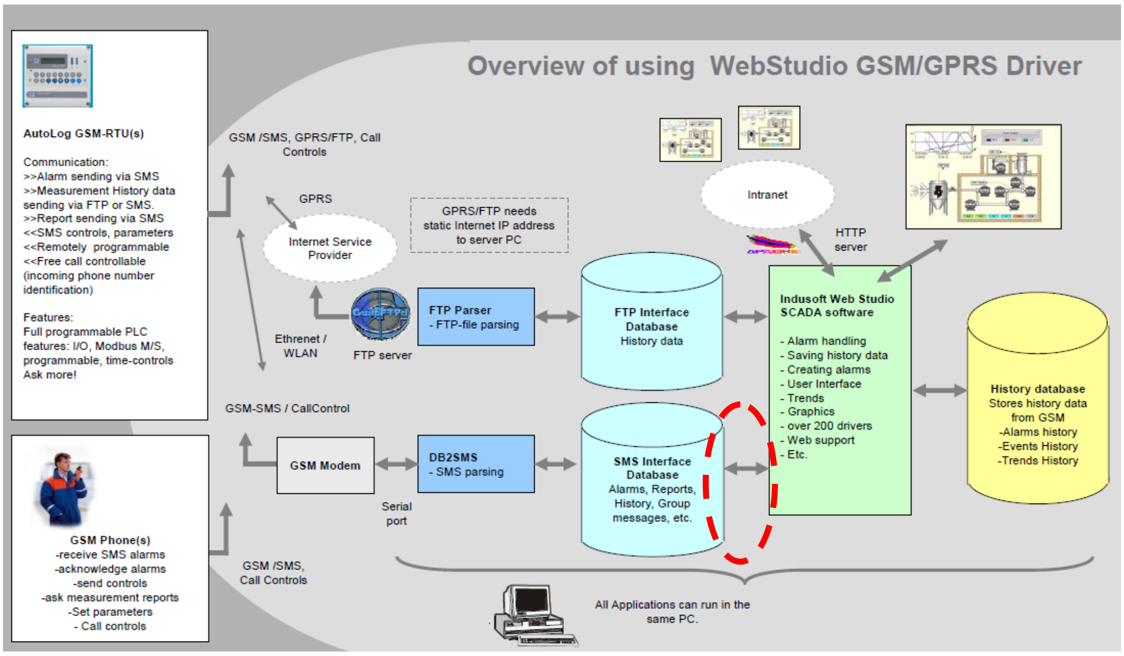
Select a driver for which you want to set up a data source.

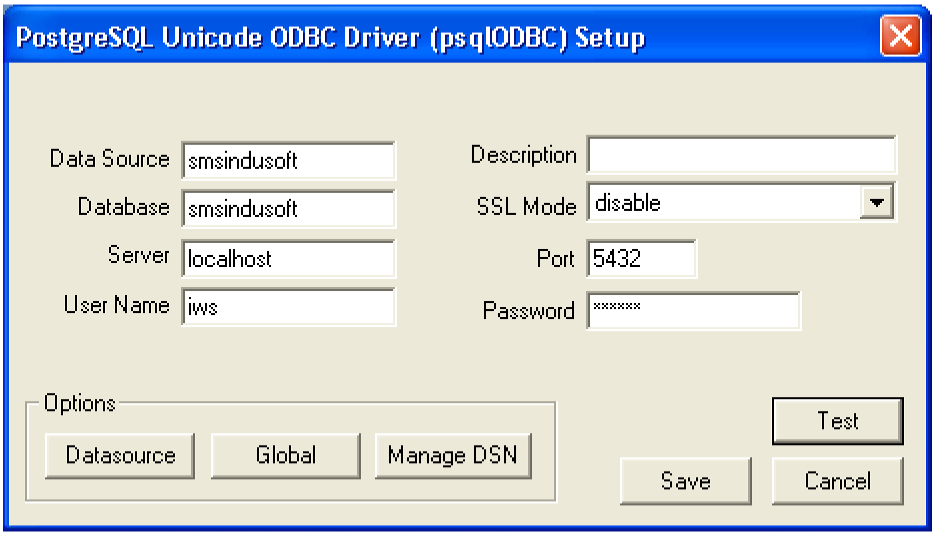
Name
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
MySQL ODBC 5.1 Driver
PostgreSQL ANSI
PostgreSQL Unicode
SQL Server

Select PostgreSQL Unicode –driver , click Finish

5.2.1 Adding ODBC DNS (Data Source Name) for DB2SMS driver

This is needed for GSM-SMS only




PostgreSQL Unicode ODBC Driver (psqlODBC) Setup

Data Source: smsindusoft

Database: smsindusoft

Server: localhost

User Name: iws

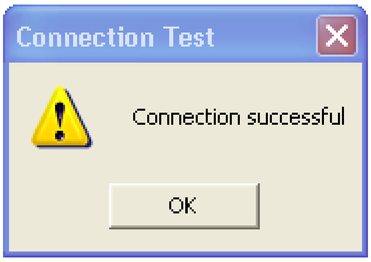
Description:

SSL Mode: disable

Port: 5432

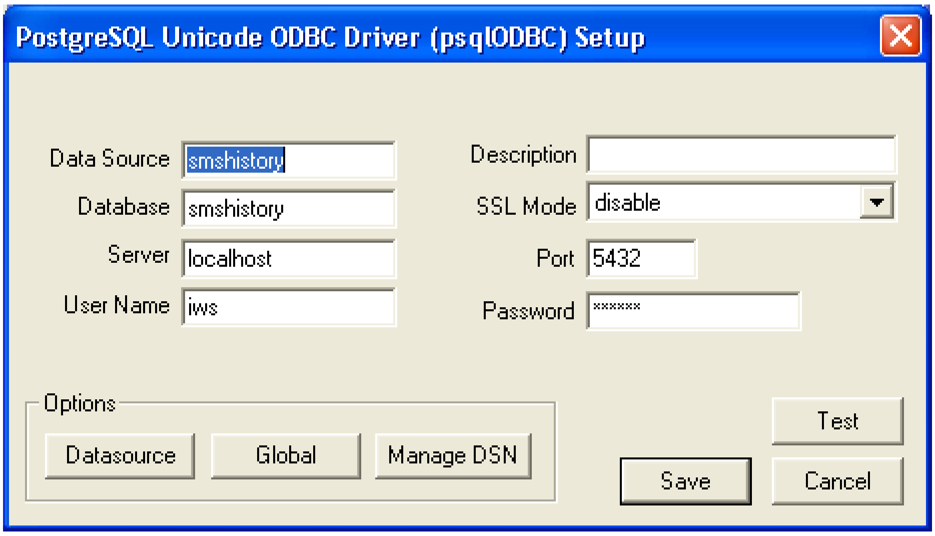
Password: ******

Make the above configuration for smsindusoft database, if you are using DB2SMS –driver. Use the same password for iws user as was configured when installing postgreSQL database.


Connection Test

Connection successful

Test connection!



PostgreSQL Unicode ODBC Driver (psqlODBC) Setup

Data Source: smshistory

Database: smshistory

Server: localhost

User Name: iws

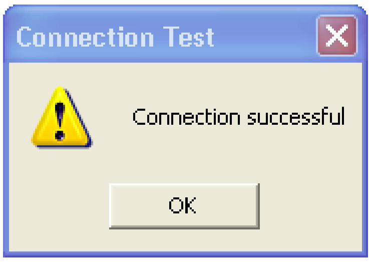
Description:

SSL Mode: disable

Port: 5432

Password: ******

Make the above configuration for smshistory database, if you are using DB2SMS – driver. Use the same password for iws user as was configured when installing postgreSQL database.



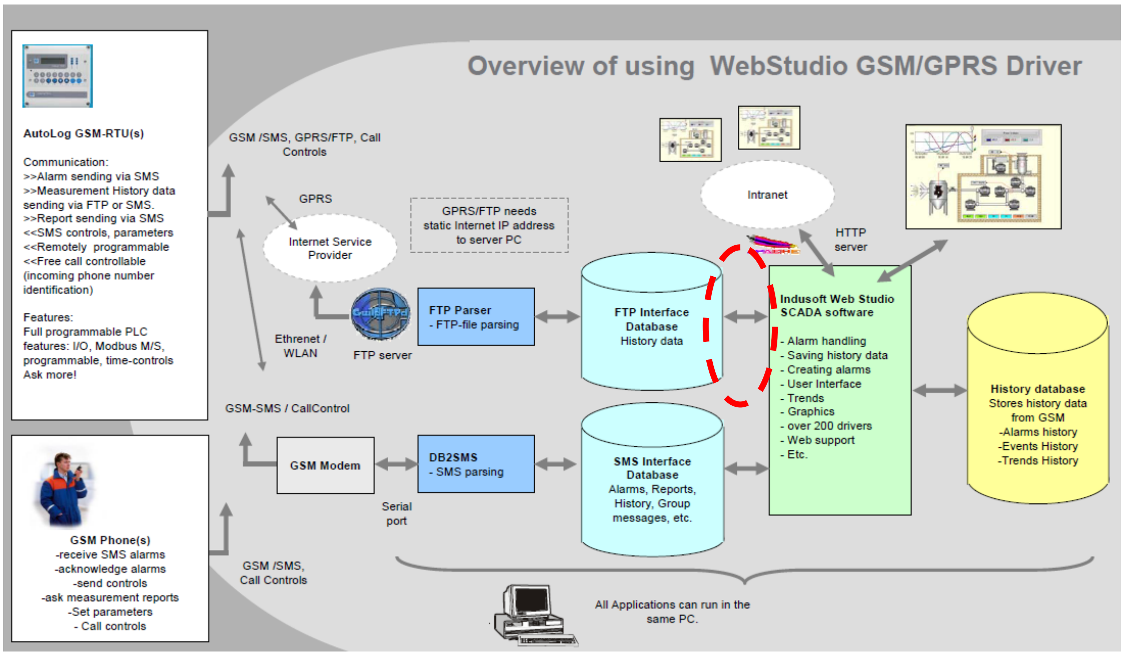
Connection Test

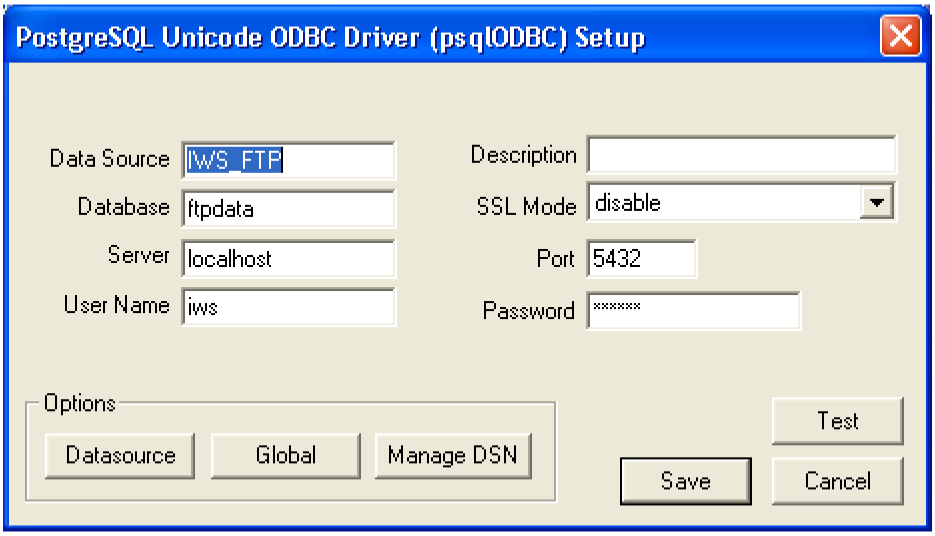
Connection successful

Test connection!

5.2.2 Adding ODBC DNS (Data Source Name) for FTP_parser driver

This is needed for GPRS/FTP only.





PostgreSQL Unicode ODBC Driver (psqlODBC) Setup

Data Source: IWS_FTP

Database: ftpdata

Server: localhost

User Name: iws

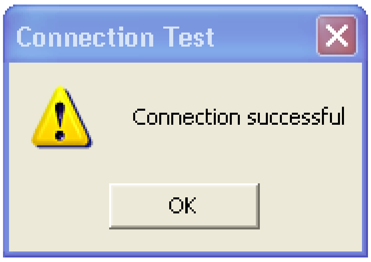
Description:

SSL Mode: disable

Port: 5432

Password: ******

Make the following configuration for ftpdata database, if you are using FTP_parser –driver. Use the same password for iws user as was configured when installing postgreSQL database.



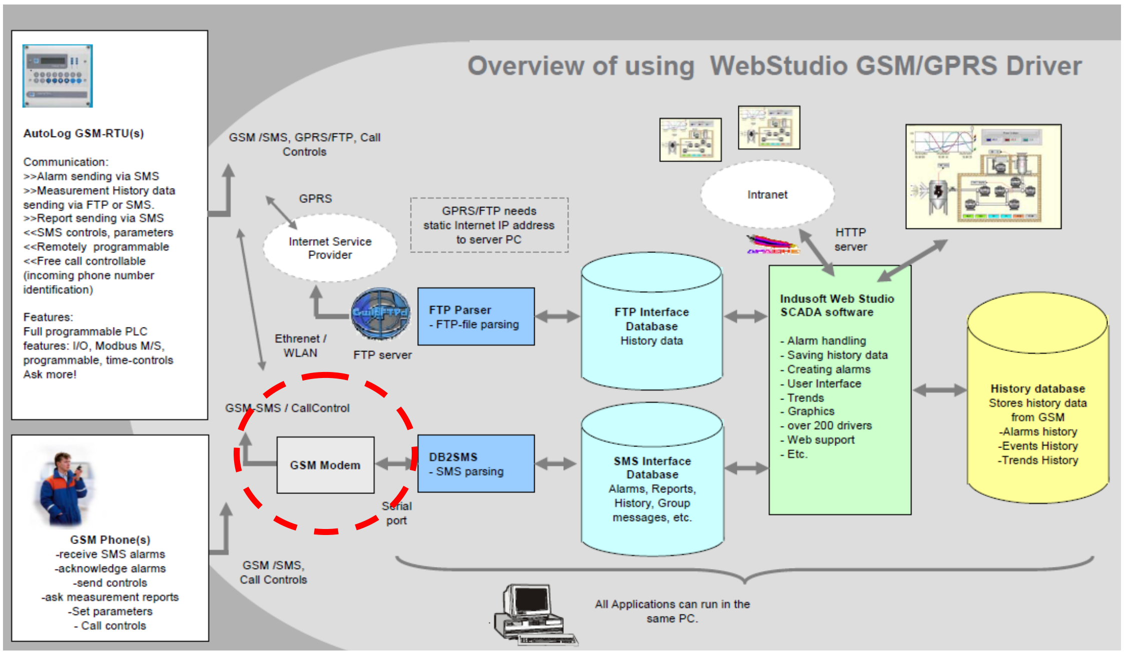
Connection Test

Connection successful

Test connection!

5.3 Connecting the GSM modem to the PC’s serial port.

This is needed for GSM-SMS only. For FTP you don’t need GPRS modem, it is using Internet connection.



Connect the GSM modem cable to PC’s serial port (check that the serial port is free so that any other program is not using it). Suggestion: connect GSM modem to PC’s serial port 1 (serial port 2 can be used for programming GSM-PLCs).

Connect other end to GSM modems serial port.

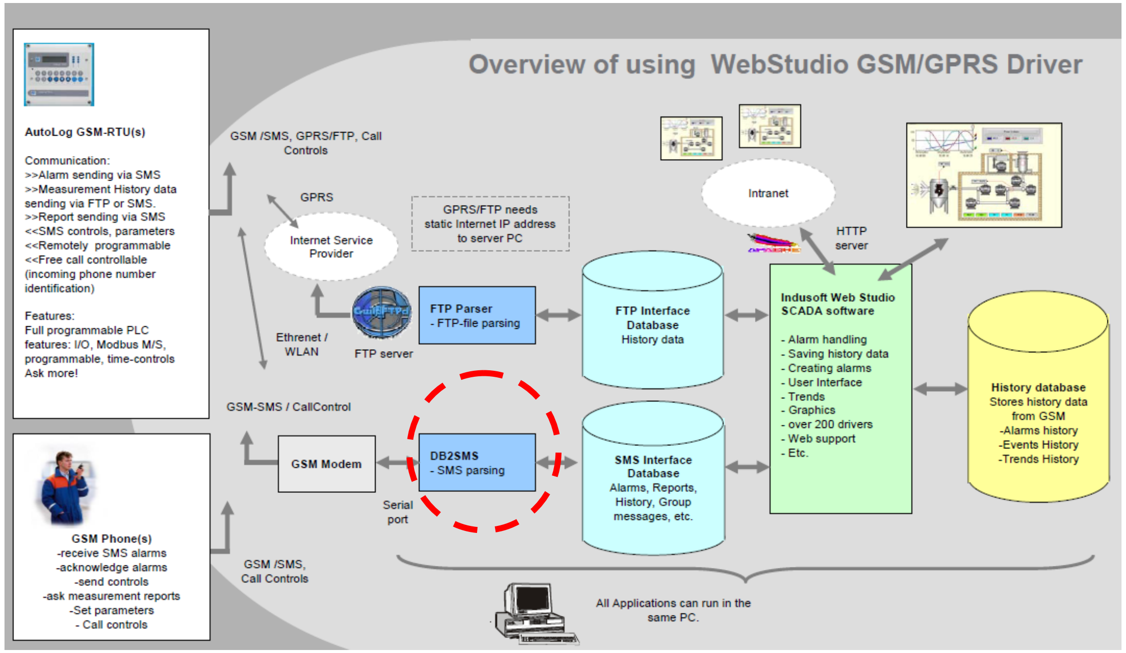
Check that the SIM card’s PIN code request is turned OFF! Move the SIM card to your mobile phone and set the PIN code request to OFF. (Look for something like Setting->Security->PIN code) You can test that it is off by opening your mobile phone. It should not ask PIN code when it starts.

Check than antenna is tightened.

Power the modem using suitable 12VDC or 24 VDC power. Look at the modem’s red indication LED, which should start blinking after initialization steadily about 2 sec intervals. If the RED led is not blinking after about 30 seconds, then try to reboot or check the manuals for possible cause.

5.4 Configuring the DB2SMS driver and GSM modem initialization test

This is needed for GSM-SMS only. For FTP you don't need GPRS modem, it is using Internet connection.



See Appendix for DB2SMS –driver interface description!

5.4.1 Configuring DB2SMS.ini file

1. Open *config.ini* with text editor. The file looks like this:

```
############################################
# Modem settings
############################################

Modem_Port='1'
```

CONFIGURATIONS

```
#Modem_PIN='0000'
Modem_Speed='9600'

# (0 = No handshaking, 1 = XOnXOff, 2 = RequestToSend, 3 = RequestToSendXOnXOff)
; Modem_Handshaking='2'
Modem_Handshaking='1'

#########################################
# Database configuration
#
# See http://www.connectionstring.com
# if you don't want use PostgreSQL or
# MySQL
#########################################

# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51
;Driver};SERVER=localhost;DATABASE=smsindusoft;USER=root;PASSWORD=securepassword;'

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=smsindusoft;Uid=iws;Pwd=ffauto;'

#########################################
# Wierd options
# These may be useful in some cases
#########################################

# Next line defines how many times modem initialization may fail before error message is given
# We've noticed that a modem gives "modem link is too weak" -alarm without any
# reason in some cases. Therefore we recommend not to comment next line.

WT_InitializationErrorMax='2'
```

(Same config.ini is also in the appendix)

CONFIGURATIONS

2. Serial port for modem is defined in the first line.
3. SIM –card’s PIN number is defined in the second line. You can leave this undefined if the *PIN code request* is configured as ‘off’ in your SIM card. Note! It is highly suggested that you turn off the PIN code request from your SIM card! Otherwise the SIM card will go to PUK mode if you have wrong PIN code.
4. Modem speed is defined in the third line. Normally this is 9600, so normally you do not need to change it.
5. Modem_Handshaking parameter is important and it should be defined correctly. If you are unsure which value to use, you can try one by one which work correctly. Look the table next table for more information!
6. The connection string defines the database connection. MySQL connection string is commented. Change the user name and password to correct ones.

    - DB2SMS uses ODBC –driver to create connection with database.
        - ConnectionString should include Server, Database, User and Password –parameters (look example file)
    - Other databases
        - Other databases are not tested with DB2SMS. In principle all the databases that uses ODBC interface should work without problems.

7. Normally you don’t need to give any other parameters to DB2SMS, but in the next table includes all the parameters that can be configured.

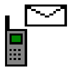
Parameter	Mandatoryn/Optional	Description
ConnectionString=’x’	Mandatory	Database connection definition
Modem_Port=’x’	Mandatory	Used serial port for GSM modem
Modem_Speed	Mandatory	Serial port speed (normally 9600)
Modem_Handshaking	Mandatory	Handshaking level 0 = No handshaking 1 = XOn / XOff 2 = Request to send 3 = Request to send and XOn / XOff
pin=’xxxx’	Optional	SIM –card’s PIN –code
license=’move’	Optional	Activates licence move state
loglevel=’x’	Optional (default 1)	Selects the level for message log: 1:Received and send messages are written to log (default). 2:The serial communication is written to log. 3: Both 1 and 2.

Config.ini Parameter explanations

Note! If your PIN code request is enabled (suggestion : PIN code request should be disabled) in your SIM card, you should not change the PIN code while the program is running. DB2SMS tries to initialize SIM card with the defined PIN code, and if the code is wrong, the SIM card probably gets locked.

5.4.2 Starting the DB2SMS program

Program can be started by double-clicking the *DB2SMS.exe* file. If your program is unlicensed, the program informs how much evaluation time you have left and when clicking the *Continue Evaluate software* –button the application is started. If the configuration of the program is correct you should see small "mobile phone" picture in the right down corner (system tray):



5.4.3 Troubleshooting - Modem initialization problems

If you have problems with initializing modem you will get some error message with some error code.

You can check that these are ok:

- You have inserted SIM card
- PIN code request is disabled / you have right PIN code in config.ini
- You have right type of cable and it is in right COM port
- Baud rate is set correctly (9600)
- Try different options 1-3 for Modem_Handshaking='1' parameter.
- Check the DB2SMS Log files and error code description – look appendix for full description of DB2SMS driver!

If the above were ok you can test that the modem is in TEXT mode and the ECHO is OFF.

Open HyperTerminal and start session with parameters COM#, 9600,8/1/N,OFF

Command	Modem should Reply
AT	OK
AT+CPIN?	OK or +CPIN: SIM PIN or +CPIN: READY If you get ERROR or +CPIN: SIM PUK, there's a Problem with the SIM.
AT+CMGF?	1 (=text mode)
ATE0	OK (sets echo off)

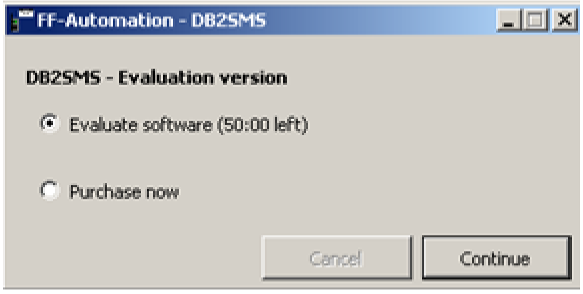
5.4.4 Closing the program

Program can be closed by double clicking the mouse's **right** button on the "mobile phone" picture in the right down corner (system tray). After that you will get confirmation message and answering 'Yes' closes the program.

5.4.5 Program license

5.4.5.1 Evaluation period

DB2SMS program can be used 50 hours evaluation time without licensing. Program informs how much time is left when it is started.



FF-Automation - DB2SMS

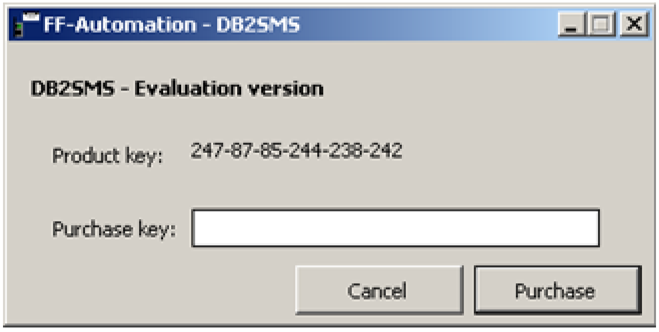
DB2SMS - Evaluation version

● Evaluate software (50:00 left)

○ Purchase now

5.4.5.2 Purchase and registration

Program can be licensed by selecting the *Purchase now* checkbox and clicking *Continue*. Program generates product key, which is unique key for PC. This key should then be send to FF-Automation (e.g. by e-mail) and FF-Automation then sends back the right *Purchase key* to activate the license. Program informs if the activation were successful.



FF-Automation - DB2SMS

DB2SMS - Evaluation version

Product key: 247-87-85-244-238-242

Purchase key:

5.4.5.3 Moving the license

The license can be moved to other PC. The original license is removed.

You should first install DB2SMS to destination PC. After that you should add the following line to the original already licensed PC's configuration file (Config.ini)

license='move'

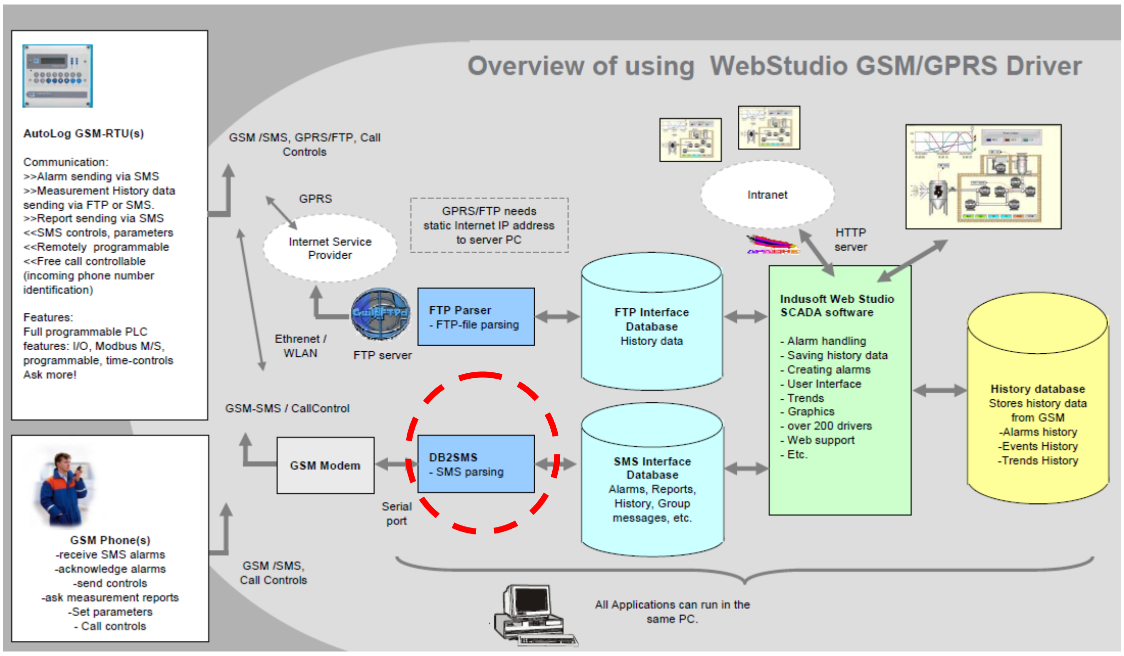
Look the previous Config .ini parameter explanations –table!

This activates the license transfer state. Write the Product key –field the destination PC's Product key and click Move. Program warns about the license transfer, if you answer 'Yes' the original license is removed and you get new Purchase key to be inserted to destination PC's Purchase key –field.

Note! Be careful when you insert the product key. If the product key is faulty, the program doesn't generate right purchase key and you will lose the license. If this happens please contact with FF-Automation. (Include the product key and the generated purchase key in your message.)

5.5 Configuring the FTP_parser –driver

This is need for GPRS/FTP only.



CONFIGURATIONS

1. Open *config.ini* with text editor. The file looks like this:

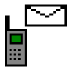
```
################################
# FTP parser configuration file #
# Updated 23.1.2008             #
################################

################################
# General configuration         #
################################

# DataDir defines the directory that holds received files
DataDir='C:\FTP'

################################
# Database connection            #
################################

# You can use MySQL or PostgreSQL with FTP parser comment the line you don't need

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL ANSI};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'
```

2. The database name, username and password was created earlier. Check that these are correct.

5.5.1 Starting the FTP_Parser –driver program

Do not start the FTP_Parser driver yet. Or if you start it remember to stop it before going to next step!

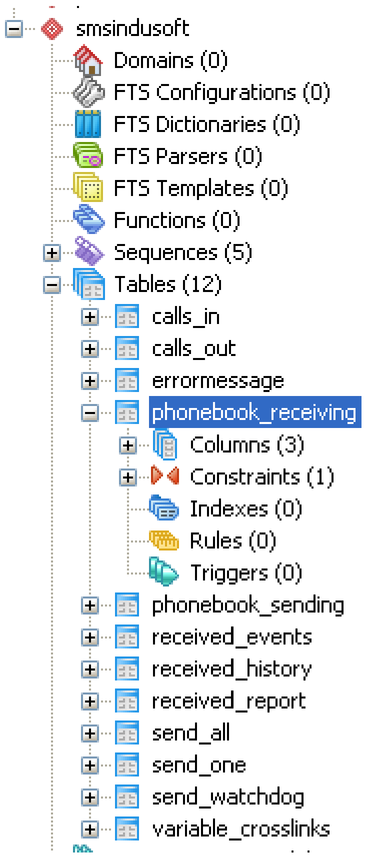
Program can be started by double-clicking the *FTPParser.exe* file. If your program is unlicensed, the program informs how much evaluation time you have left and when clicking the *Continue Evaluate software* –button the application is started. If the configuration of the program is correct you should see small “mobile phone” picture in the right down corner (system tray):



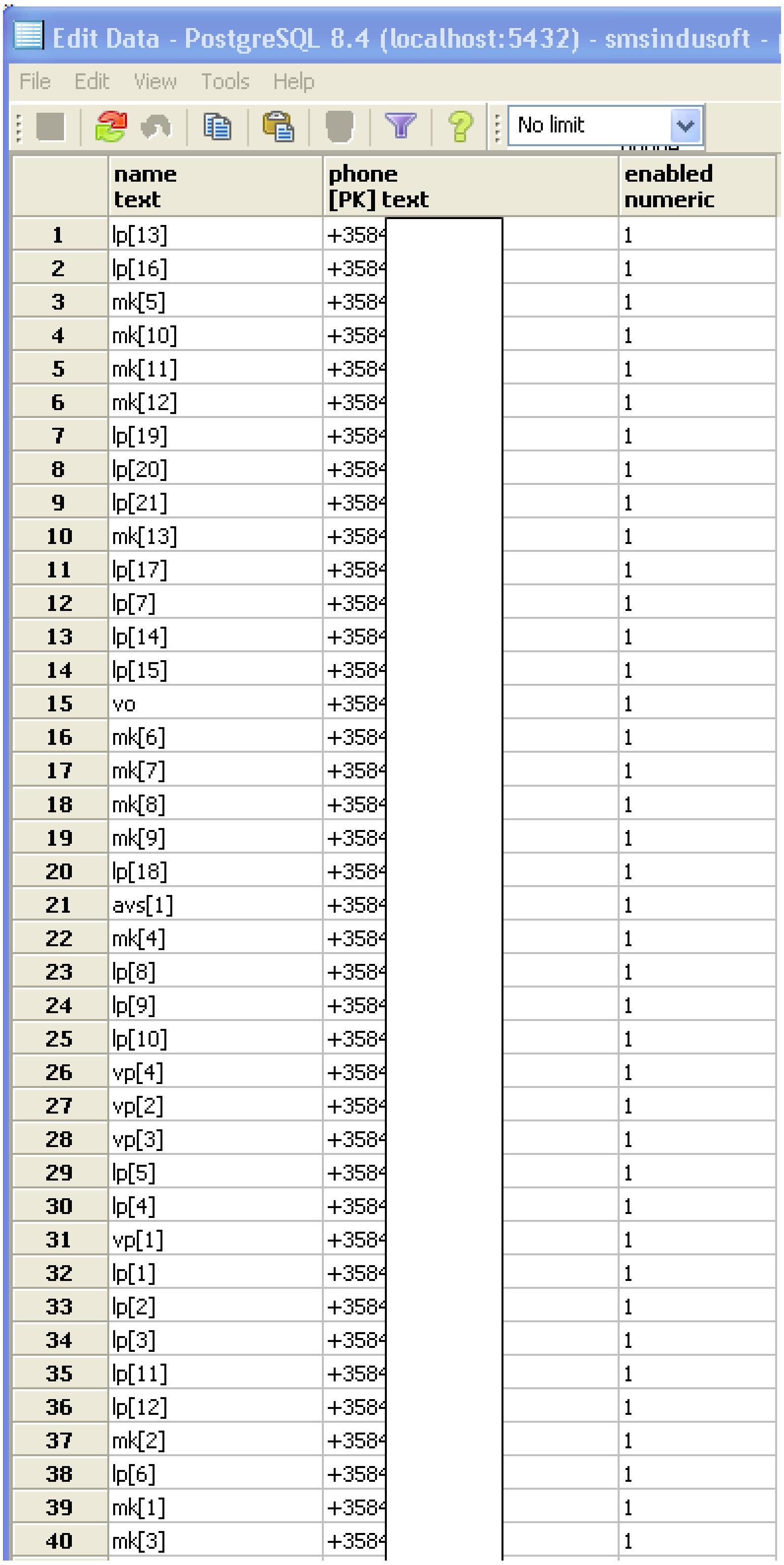
5.5.2 Closing the FTP_parser program

Program can be closed by double clicking the mouse’s **right** button on the “mobile phone” picture in the right down corner (system tray). After that you will get confirmation message and answering ‘Yes’ closes the program.

6 Configuring PostgreSQL smsindusoft database's tables for DB2SMS and IWS application.



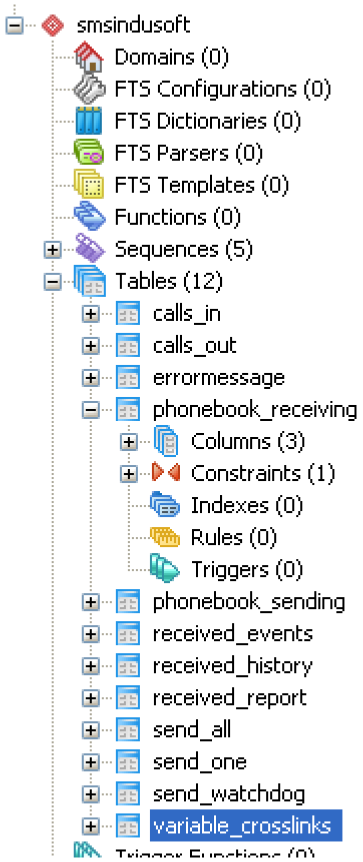
Open pgAdminIII and open smsindusoft database. Right click the phone book receiving and select View data->View all rows



	name text	phone [PK] text	enabled numeric
1	lp[13]	+3584[redacted]	1
2	lp[16]	+3584[redacted]	1
3	mk[5]	+3584[redacted]	1
4	mk[10]	+3584[redacted]	1
5	mk[11]	+3584[redacted]	1
6	mk[12]	+3584[redacted]	1
7	lp[19]	+3584[redacted]	1
8	lp[20]	+3584[redacted]	1
9	lp[21]	+3584[redacted]	1
10	mk[13]	+3584[redacted]	1
11	lp[17]	+3584[redacted]	1
12	lp[7]	+3584[redacted]	1
13	lp[14]	+3584[redacted]	1
14	lp[15]	+3584[redacted]	1
15	vo	+3584[redacted]	1
16	mk[6]	+3584[redacted]	1
17	mk[7]	+3584[redacted]	1
18	mk[8]	+3584[redacted]	1
19	mk[9]	+3584[redacted]	1
20	lp[18]	+3584[redacted]	1
21	avs[1]	+3584[redacted]	1
22	mk[4]	+3584[redacted]	1
23	lp[8]	+3584[redacted]	1
24	lp[9]	+3584[redacted]	1
25	lp[10]	+3584[redacted]	1
26	vp[4]	+3584[redacted]	1
27	vp[2]	+3584[redacted]	1
28	vp[3]	+3584[redacted]	1
29	lp[5]	+3584[redacted]	1
30	lp[4]	+3584[redacted]	1
31	vp[1]	+3584[redacted]	1
32	lp[1]	+3584[redacted]	1
33	lp[2]	+3584[redacted]	1
34	lp[3]	+3584[redacted]	1
35	lp[11]	+3584[redacted]	1
36	lp[12]	+3584[redacted]	1
37	mk[2]	+3584[redacted]	1
38	lp[6]	+3584[redacted]	1
39	mk[1]	+3584[redacted]	1
40	mk[3]	+3584[redacted]	1

Add the GSM-PLC's SIM cards phone number to the list with land area code, no spaces. Insert value 1 to enable column. Name column is used to link the data between variable_crosslinks table which we will configure next.

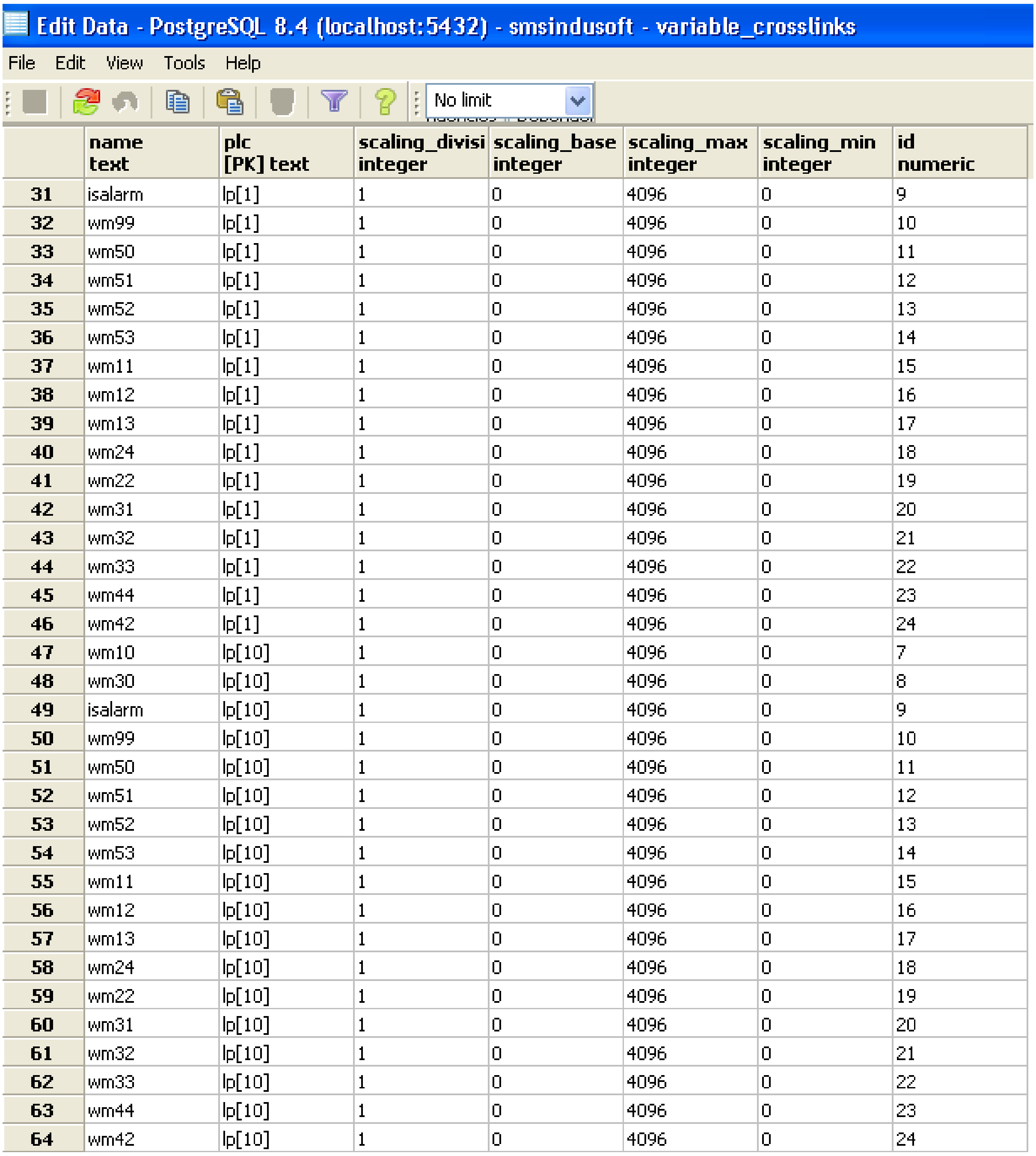
CONFIGURATIONS



```
smsindusoft
  Domains (0)
  FTS Configurations (0)
  FTS Dictionaries (0)
  FTS Parsers (0)
  FTS Templates (0)
  Functions (0)
  Sequences (5)
  Tables (12)
    calls_in
    calls_out
    errormessage
    phonebook_receiving
      Columns (3)
      Constraints (1)
      Indexes (0)
      Rules (0)
      Triggers (0)
    phonebook_sending
    received_events
    received_history
    received_report
    send_all
    send_one
    send_watchdog
    variable_crosslinks
  Trigger Functions (0)
```

Right click the variable_crosslinks table and select View data->View all rows

CONFIGURATIONS


Edit Data - PostgreSQL 8.4 (localhost:5432) - smsindusoft - variable_crosslinks
File
Edit
View
Tools
Help
No limit

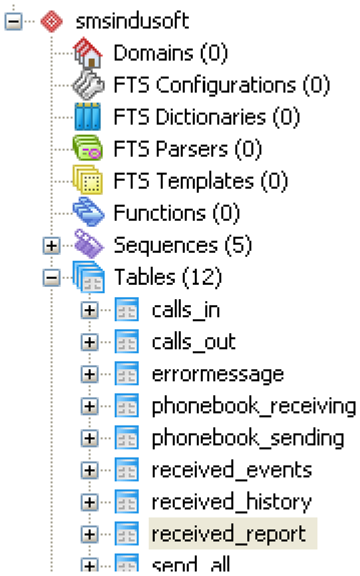
	name text	plc [PK] text	scaling_divisi integer	scaling_base integer	scaling_max integer	scaling_min integer	id numeric
31	isalarm	lp[1]	1	0	4096	0	9
32	wm99	lp[1]	1	0	4096	0	10
33	wm50	lp[1]	1	0	4096	0	11
34	wm51	lp[1]	1	0	4096	0	12
35	wm52	lp[1]	1	0	4096	0	13
36	wm53	lp[1]	1	0	4096	0	14
37	wm11	lp[1]	1	0	4096	0	15
38	wm12	lp[1]	1	0	4096	0	16
39	wm13	lp[1]	1	0	4096	0	17
40	wm24	lp[1]	1	0	4096	0	18
41	wm22	lp[1]	1	0	4096	0	19
42	wm31	lp[1]	1	0	4096	0	20
43	wm32	lp[1]	1	0	4096	0	21
44	wm33	lp[1]	1	0	4096	0	22
45	wm44	lp[1]	1	0	4096	0	23
46	wm42	lp[1]	1	0	4096	0	24
47	wm10	lp[10]	1	0	4096	0	7
48	wm30	lp[10]	1	0	4096	0	8
49	isalarm	lp[10]	1	0	4096	0	9
50	wm99	lp[10]	1	0	4096	0	10
51	wm50	lp[10]	1	0	4096	0	11
52	wm51	lp[10]	1	0	4096	0	12
53	wm52	lp[10]	1	0	4096	0	13
54	wm53	lp[10]	1	0	4096	0	14
55	wm11	lp[10]	1	0	4096	0	15
56	wm12	lp[10]	1	0	4096	0	16
57	wm13	lp[10]	1	0	4096	0	17
58	wm24	lp[10]	1	0	4096	0	18
59	wm22	lp[10]	1	0	4096	0	19
60	wm31	lp[10]	1	0	4096	0	20
61	wm32	lp[10]	1	0	4096	0	21
62	wm33	lp[10]	1	0	4096	0	22
63	wm44	lp[10]	1	0	4096	0	23
64	wm42	lp[10]	1	0	4096	0	24

Here’s two example GSM-RTUs lp[1] and lp[10]
Id = the incoming SMS messages variable number
“D9,,0,%AI0@7,%WM91” ; this is part of the programming line in GSMProgrammer. The first parameter D9 indicates the first ID number so that the 0 parameter is id=9, and AI0=id10 etc. Id =9 parameter can be used freely e.g. to identify the message source e.g. if it was generated by digital input or analog input or clock etc. It is just an example so it is not mandatory.

The name is the variable name in the Indusoft web studio which will be combined with plc so that e.g. lp[1].wm99 gets the value from the id=9 and plc phone number from phonebook_receiving.

If you are configuring many GSM-PLC’s you can use PostGreSQL’s command prompt to insert values to these tables.

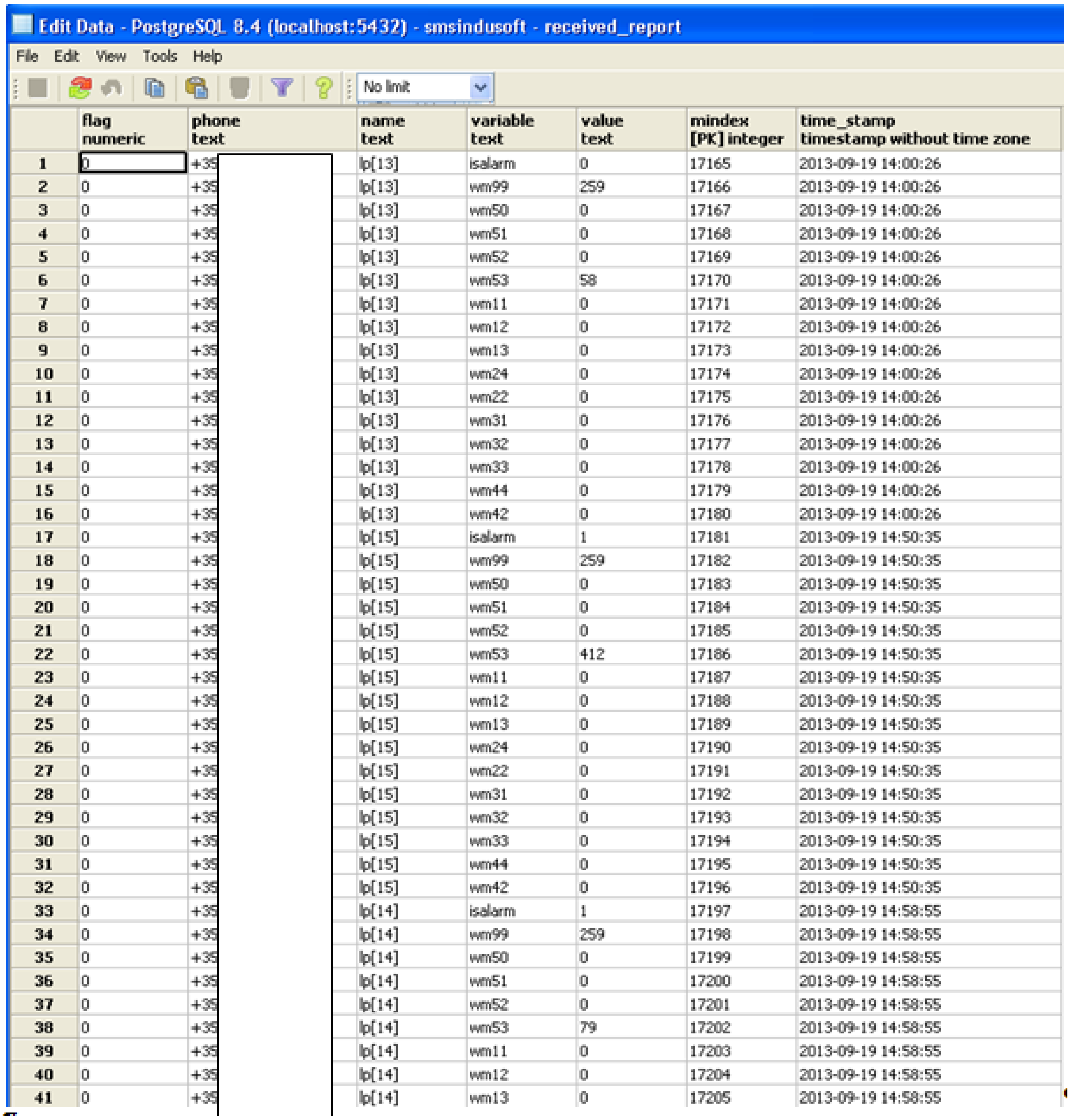
Testing the configurations:



smsindusoft
Domains (0)
FTS Configurations (0)
FTS Dictionaries (0)
FTS Parsers (0)
FTS Templates (0)
Functions (0)
Sequences (5)
Tables (12)
calls_in
calls_out
errormessage
phonebook_receiving
phonebook_sending
received_events
received_history
received_report
send_all

The data from GSM-RTU will be parsed automatically to “received_report” table.

Open the table by right clicking and selecting view data->view all rows


Edit Data - PostgreSQL 8.4 (localhost:5432) - smsindusoft - received_report
File Edit View Tools Help
No limit

	flag	phone	name	variable	value	mindex	time_stamp
	numeric	text	text	text	text	[PK] integer	timestamp without time zone
1	0	+3[redacted]	lp[13]	isalarm	0	17165	2013-09-19 14:00:26
2	0	+3[redacted]	lp[13]	wm99	259	17166	2013-09-19 14:00:26
3	0	+3[redacted]	lp[13]	wm50	0	17167	2013-09-19 14:00:26
4	0	+3[redacted]	lp[13]	wm51	0	17168	2013-09-19 14:00:26
5	0	+3[redacted]	lp[13]	wm52	0	17169	2013-09-19 14:00:26
6	0	+3[redacted]	lp[13]	wm53	58	17170	2013-09-19 14:00:26
7	0	+3[redacted]	lp[13]	wm11	0	17171	2013-09-19 14:00:26
8	0	+3[redacted]	lp[13]	wm12	0	17172	2013-09-19 14:00:26
9	0	+3[redacted]	lp[13]	wm13	0	17173	2013-09-19 14:00:26
10	0	+3[redacted]	lp[13]	wm24	0	17174	2013-09-19 14:00:26
11	0	+3[redacted]	lp[13]	wm22	0	17175	2013-09-19 14:00:26
12	0	+3[redacted]	lp[13]	wm31	0	17176	2013-09-19 14:00:26
13	0	+3[redacted]	lp[13]	wm32	0	17177	2013-09-19 14:00:26
14	0	+3[redacted]	lp[13]	wm33	0	17178	2013-09-19 14:00:26
15	0	+3[redacted]	lp[13]	wm44	0	17179	2013-09-19 14:00:26
16	0	+3[redacted]	lp[13]	wm42	0	17180	2013-09-19 14:00:26
17	0	+3[redacted]	lp[15]	isalarm	1	17181	2013-09-19 14:50:35
18	0	+3[redacted]	lp[15]	wm99	259	17182	2013-09-19 14:50:35
19	0	+3[redacted]	lp[15]	wm50	0	17183	2013-09-19 14:50:35
20	0	+3[redacted]	lp[15]	wm51	0	17184	2013-09-19 14:50:35
21	0	+3[redacted]	lp[15]	wm52	0	17185	2013-09-19 14:50:35
22	0	+3[redacted]	lp[15]	wm53	412	17186	2013-09-19 14:50:35
23	0	+3[redacted]	lp[15]	wm11	0	17187	2013-09-19 14:50:35
24	0	+3[redacted]	lp[15]	wm12	0	17188	2013-09-19 14:50:35
25	0	+3[redacted]	lp[15]	wm13	0	17189	2013-09-19 14:50:35
26	0	+3[redacted]	lp[15]	wm24	0	17190	2013-09-19 14:50:35
27	0	+3[redacted]	lp[15]	wm22	0	17191	2013-09-19 14:50:35
28	0	+3[redacted]	lp[15]	wm31	0	17192	2013-09-19 14:50:35
29	0	+3[redacted]	lp[15]	wm32	0	17193	2013-09-19 14:50:35
30	0	+3[redacted]	lp[15]	wm33	0	17194	2013-09-19 14:50:35
31	0	+3[redacted]	lp[15]	wm44	0	17195	2013-09-19 14:50:35
32	0	+3[redacted]	lp[15]	wm42	0	17196	2013-09-19 14:50:35
33	0	+3[redacted]	lp[14]	isalarm	1	17197	2013-09-19 14:58:55
34	0	+3[redacted]	lp[14]	wm99	259	17198	2013-09-19 14:58:55
35	0	+3[redacted]	lp[14]	wm50	0	17199	2013-09-19 14:58:55
36	0	+3[redacted]	lp[14]	wm51	0	17200	2013-09-19 14:58:55
37	0	+3[redacted]	lp[14]	wm52	0	17201	2013-09-19 14:58:55
38	0	+3[redacted]	lp[14]	wm53	79	17202	2013-09-19 14:58:55
39	0	+3[redacted]	lp[14]	wm11	0	17203	2013-09-19 14:58:55
40	0	+3[redacted]	lp[14]	wm12	0	17204	2013-09-19 14:58:55
41	0	+3[redacted]	lp[14]	wm13	0	17205	2013-09-19 14:58:55

In this table you can see the data from gsm-plcs. It is filled automatically when you receive data from GSM-PLCs.

Flag number (value 0) indicates that the data is read to the Indusoft. Value 1 indicates that it is not yet read to Indusoft.

Time stamp is read from the SMS message's internal timestamp field.

When you get data to this table you can move to Indusoft web studio and try to read data from this table to the Indusoft variables. See the IWS_demo template application for more info! Open the IWS's ODBC task which is used to read data from this table.

See also the Math tasks Report_Receiver* codes and scheduler codes.

After the variables are read to Indusoft tags, then it can be stored to PostGreSQL's smshistory database so that all the GSM-RTUs have own table for storing history.

7 Testing the GSM_to_Indusoft_FTP_DEMO

This chapter descripes, step-by-step, how you can set up test system for the FTP DEMO application, which has the following fuctionality:

- GSM-PLC sends FTP files over GPRS connection to your PC's FTP Server.
- FTP_Parser –driver parses the FTP file to PosGreSQL's ftpdata –database.
- Indusoft Web Studio reads the dat from database using ODBC task and shows the data in variety of interfaces.

FTP DEMO –application files and other related DEMO application files are included to "driver package". If you don't have these files please contact FF-Automation!

7.1 Setting up GSM-PLC to send FTP files via direct cable connection

Look GSM-PLC service manual for basic instructions! Look also GSM demo application for sending FTP files!

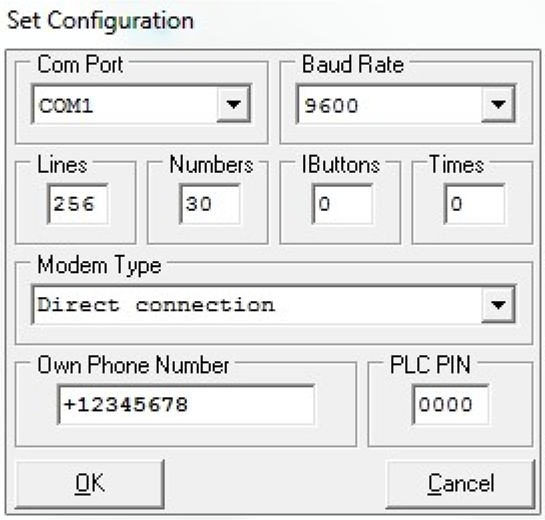
Open the GsmProgrammer
Open the GSM_to_Indusoft_FTP_Demo.SMS project

Set GSM-PLC's DIP 4 position = ON position and reboot the GSM-PLC (disables FTP transfer in GSM-4,8 and 16) not in GSM-20. (for GSM.20 take the cable of from the GSM modem)

Check that the FTP_Parser –program is stopped. If not right click on this icon on the system tray and stop it.

Connect programming cable to GSM-PLC's serial port 1.

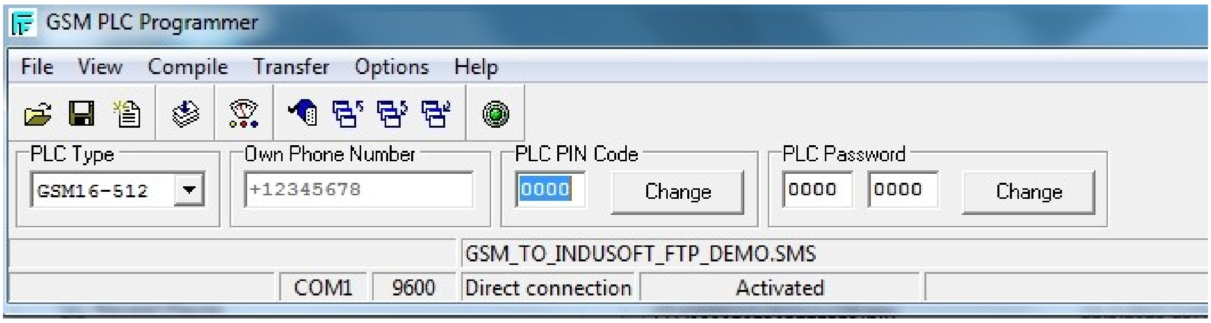
Check the following settings:
Transfer->Config



Set Configuration

Com Port: COM1
Baud Rate: 9600
Lines: 256
Numbers: 30
IButtons: 0
Times: 0
Modem Type: Direct connection
Own Phone Number: +12345678
PLC PIN: 0000

CONFIGURATIONS



PLC Type: GSM16-512

Own Phone Number: +12345678

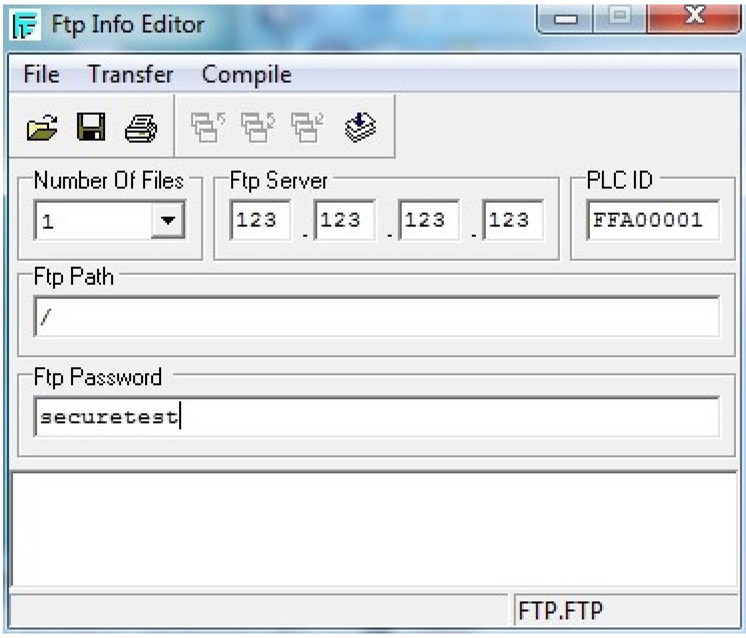
PLC PIN Code: 0000

PLC Password: 0000 0000

GSM_TO_INDUSOFT_FTP_DEMO.SMS

COM1 | 9600 | Direct connection | Activated

Check that the connection status is :Activated. It may take max. about 1 minute to connect.



Number Of Files: 1

Ftp Server: 123 . 123 . 123 . 123

PLC ID: FFA00001

Ftp Path: /

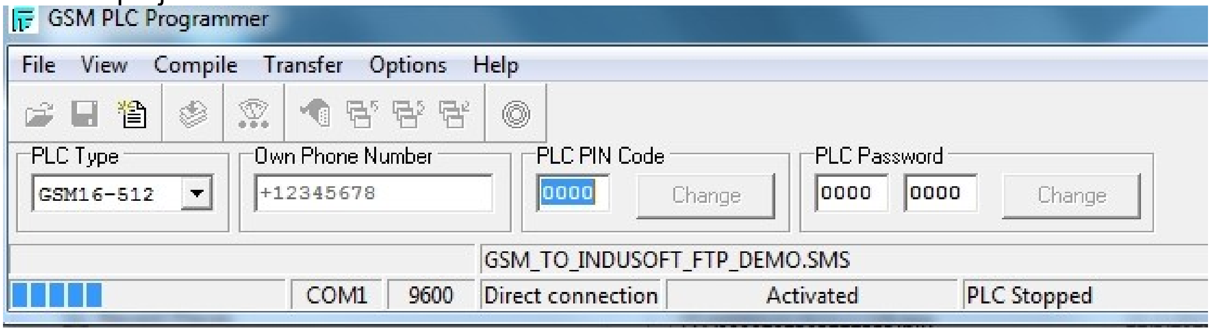
Ftp Password: securetest

FTP.FTP

View->FTP

Compile->Compile

Transfer project to GSM-PLC



PLC Type: GSM16-512

Own Phone Number: +12345678

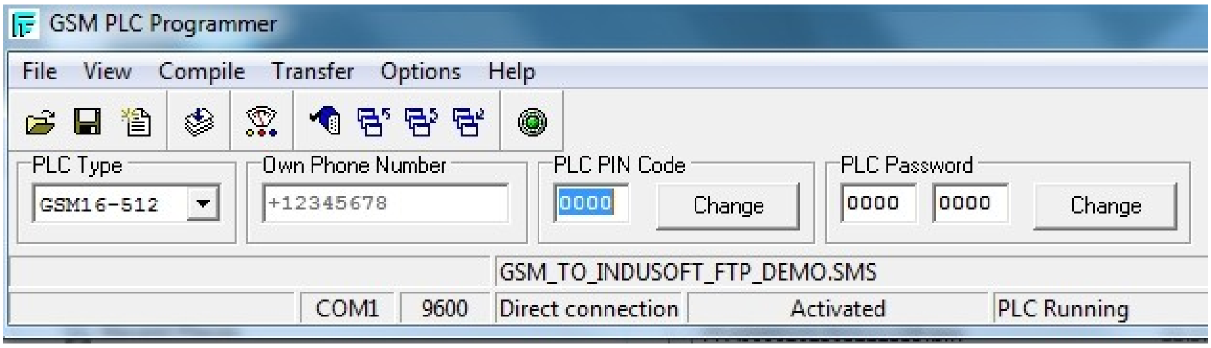
PLC PIN Code: 0000

PLC Password: 0000 0000

GSM_TO_INDUSOFT_FTP_DEMO.SMS

COM1 | 9600 | Direct connection | Activated | PLC Stopped

When you transfer the project the PLC status becomes : PLC Stopped. After transfer you need to click green RUN button.



PLC Type: GSM16-512

Own Phone Number: +12345678

PLC PIN Code: 0000

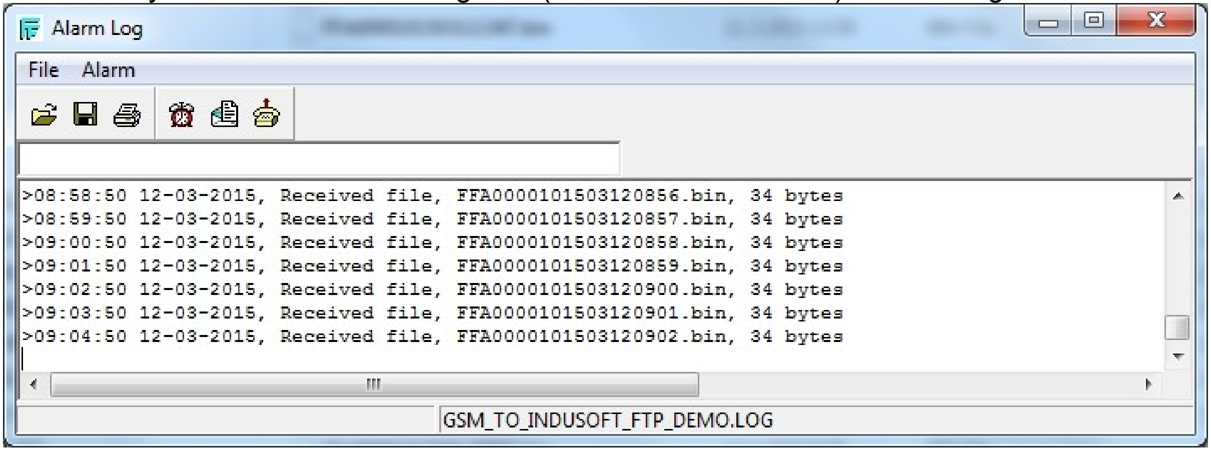
PLC Password: 0000 0000

GSM_TO_INDUSOFT_FTP_DEMO.SMS

COM1 | 9600 | Direct connection | Activated | PLC Running

Now the status is PLC Running

Check that you will receive incoming files (1 new file / 60 second) in alarm log view

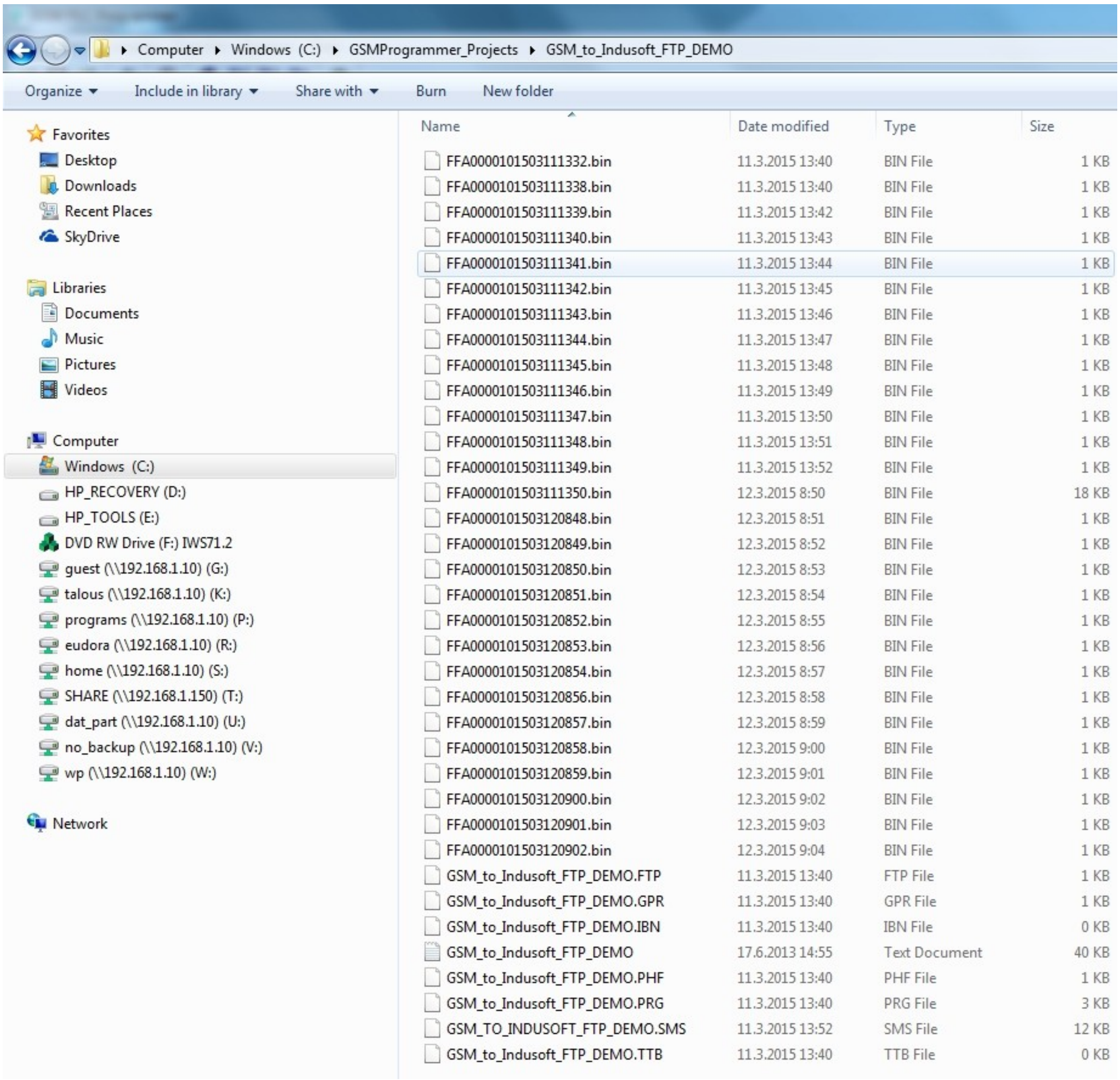


```
>08:58:50 12-03-2015, Received file, FFA0000101503120856.bin, 34 bytes
>08:59:50 12-03-2015, Received file, FFA0000101503120857.bin, 34 bytes
>09:00:50 12-03-2015, Received file, FFA0000101503120858.bin, 34 bytes
>09:01:50 12-03-2015, Received file, FFA0000101503120859.bin, 34 bytes
>09:02:50 12-03-2015, Received file, FFA0000101503120900.bin, 34 bytes
>09:03:50 12-03-2015, Received file, FFA0000101503120901.bin, 34 bytes
>09:04:50 12-03-2015, Received file, FFA0000101503120902.bin, 34 bytes
```

GSM_TO_INDUSOFT_FTP_DEMO.LOG

CONFIGURATIONS

Check that you will receive incoming files in the GsmProgrammer project’s directory.



Computer ▸ Windows (C:) ▸ GSMProgrammer_Projects ▸ GSM_to_Indusoft_FTP_DEMO
Organize ▾
Include in library ▾
Share with ▾
Burn
New folder
Favorites
Desktop
Downloads
Recent Places
SkyDrive
Libraries
Documents
Music
Pictures
Videos
Computer
Windows (C:)
HP_RECOVERY (D:)
HP_TOOLS (E:)
DVD RW Drive (F:) IWS71.2
guest (\\192.168.1.10) (G:)
talous (\\192.168.1.10) (K:)
programs (\\192.168.1.10) (P:)
eudora (\\192.168.1.10) (R:)
home (\\192.168.1.10) (S:)
SHARE (\\192.168.1.150) (T:)
dat_part (\\192.168.1.10) (U:)
no_backup (\\192.168.1.10) (V:)
wp (\\192.168.1.10) (W:)
Network

Name	Date modified	Type	Size
FFA0000101503111332.bin	11.3.2015 13:40	BIN File	1 KB
FFA0000101503111338.bin	11.3.2015 13:40	BIN File	1 KB
FFA0000101503111339.bin	11.3.2015 13:42	BIN File	1 KB
FFA0000101503111340.bin	11.3.2015 13:43	BIN File	1 KB
FFA0000101503111341.bin	11.3.2015 13:44	BIN File	1 KB
FFA0000101503111342.bin	11.3.2015 13:45	BIN File	1 KB
FFA0000101503111343.bin	11.3.2015 13:46	BIN File	1 KB
FFA0000101503111344.bin	11.3.2015 13:47	BIN File	1 KB
FFA0000101503111345.bin	11.3.2015 13:48	BIN File	1 KB
FFA0000101503111346.bin	11.3.2015 13:49	BIN File	1 KB
FFA0000101503111347.bin	11.3.2015 13:50	BIN File	1 KB
FFA0000101503111348.bin	11.3.2015 13:51	BIN File	1 KB
FFA0000101503111349.bin	11.3.2015 13:52	BIN File	1 KB
FFA0000101503111350.bin	12.3.2015 8:50	BIN File	18 KB
FFA0000101503120848.bin	12.3.2015 8:51	BIN File	1 KB
FFA0000101503120849.bin	12.3.2015 8:52	BIN File	1 KB
FFA0000101503120850.bin	12.3.2015 8:53	BIN File	1 KB
FFA0000101503120851.bin	12.3.2015 8:54	BIN File	1 KB
FFA0000101503120852.bin	12.3.2015 8:55	BIN File	1 KB
FFA0000101503120853.bin	12.3.2015 8:56	BIN File	1 KB
FFA0000101503120854.bin	12.3.2015 8:57	BIN File	1 KB
FFA0000101503120856.bin	12.3.2015 8:58	BIN File	1 KB
FFA0000101503120857.bin	12.3.2015 8:59	BIN File	1 KB
FFA0000101503120858.bin	12.3.2015 9:00	BIN File	1 KB
FFA0000101503120859.bin	12.3.2015 9:01	BIN File	1 KB
FFA0000101503120900.bin	12.3.2015 9:02	BIN File	1 KB
FFA0000101503120901.bin	12.3.2015 9:03	BIN File	1 KB
FFA0000101503120902.bin	12.3.2015 9:04	BIN File	1 KB
GSM_to_Indusoft_FTP_DEMO.FTP	11.3.2015 13:40	FTP File	1 KB
GSM_to_Indusoft_FTP_DEMO.GPR	11.3.2015 13:40	GPR File	1 KB
GSM_to_Indusoft_FTP_DEMO.IBN	11.3.2015 13:40	IBN File	0 KB
GSM_to_Indusoft_FTP_DEMO	17.6.2013 14:55	Text Document	40 KB
GSM_to_Indusoft_FTP_DEMO.PHF	11.3.2015 13:40	PHF File	1 KB
GSM_to_Indusoft_FTP_DEMO.PRG	11.3.2015 13:40	PRG File	3 KB
GSM_TO_INDUSOFT_FTP_DEMO.SMS	11.3.2015 13:52	SMS File	12 KB
GSM_to_Indusoft_FTP_DEMO.TTB	11.3.2015 13:40	TTB File	0 KB

Test was successful!

Troubleshooting: If you see that the files are received but you cannot see the files in the project directory, check the security settings of the project folder. In Windows 7 the Program Files directory is by default set as read only. You can make new project directory to e.g. C:\GSMProgrammer_Projects\ and copy the

View->Source code

TESTING FTP DEMO

7.1.1 Here’s one GSM-PLC GSMProgrammer source code version for sending FTP files.

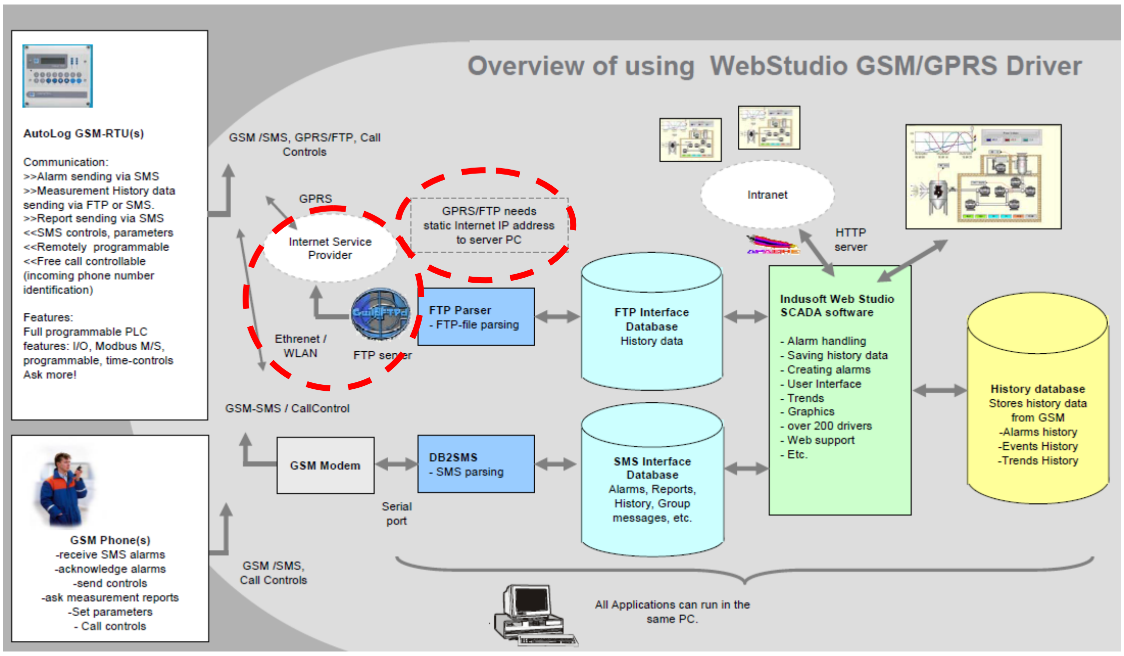
```
;Program: FTP program
;Version: 1.1
;Date   : 23.5.2013
;Author : Antti Moijanen v.1.1 / (Jouni Paavonen v1.0 base code)

;********* FTP LOG & SEND PARAMETERS **********
;message in format "$FTP=5=60", where 5 defines log interval in minutes and 60 defiens the tx interval (in minutes)
;'($FTP)' M0 WM35=WM0 WM36=WM1
;'M0' "Log interval= %WM35 seconds, Tx interval = %WM36 seconds" 254
;Check for incorrect values

;'M100#!M100' WM35=30 ;seconds
;'WM35=0' WM35=1
;'WM36=0' WM36=5


'M100#!M100' WM20=AI0 WM21=AI1 WM22=AI2 WM23=AI3;executed always, moves RAW Analog input values to WM20-23

;********* FTP LOGGING & Transfer ***********

;Optional clause: If some of the output status changes, log & send, uncomment the next line to activate it
;'DO0=1#DO0=0#DO1=1#DO1=0#DO2=1#DO3=1#DO3=0' M62=1              ;log & send on event

'P1' WM30+1             ;increment logging timer and transfer timer once/second (P1)
'P1' WM100+1                    ;increment communication watchdog once/second (P1)
'WM100=65535' WM100=1000        ;resets communication watchdog to 1000 when reached 65535

;**** LOGGING ****
'WM30=25' M60                           ;timer baser logging (M60)
'M60=1' "%WM20,%WM21,%WM22,%WM23,%WM100,%RO98" 245

;**** TRANSFER ****
'WM30=30' M61 WM30=0                            ;Timer based Transmit (M61)
'M61=1' RO38=8 RO37=1 ;                         ; RO37 & RO38 are special FTP sending registers, look GSM-PLC User
Manual!

;Backup SMS sending if enabled
```

```
;Send if error in FTP transfer. Exclude file empty error (130)
;'RO37>2&!(RO37=130)' "WM20,%WM21,%WM22,%WM23,%WM100,%RO98" 20 ; sends SMS to phone number 20 defined in phone book!

; Note when testing this application use direct cable connection and DIP 4 position = ON, before testing the real
GPRS transfer.
; Look incoming files in alarm log view and the GsmProgrammer program root directory.

See appendix for little bit more advanced version of this program!
```

7.2 Testing the FTP file transfer over GPRS to FTP server

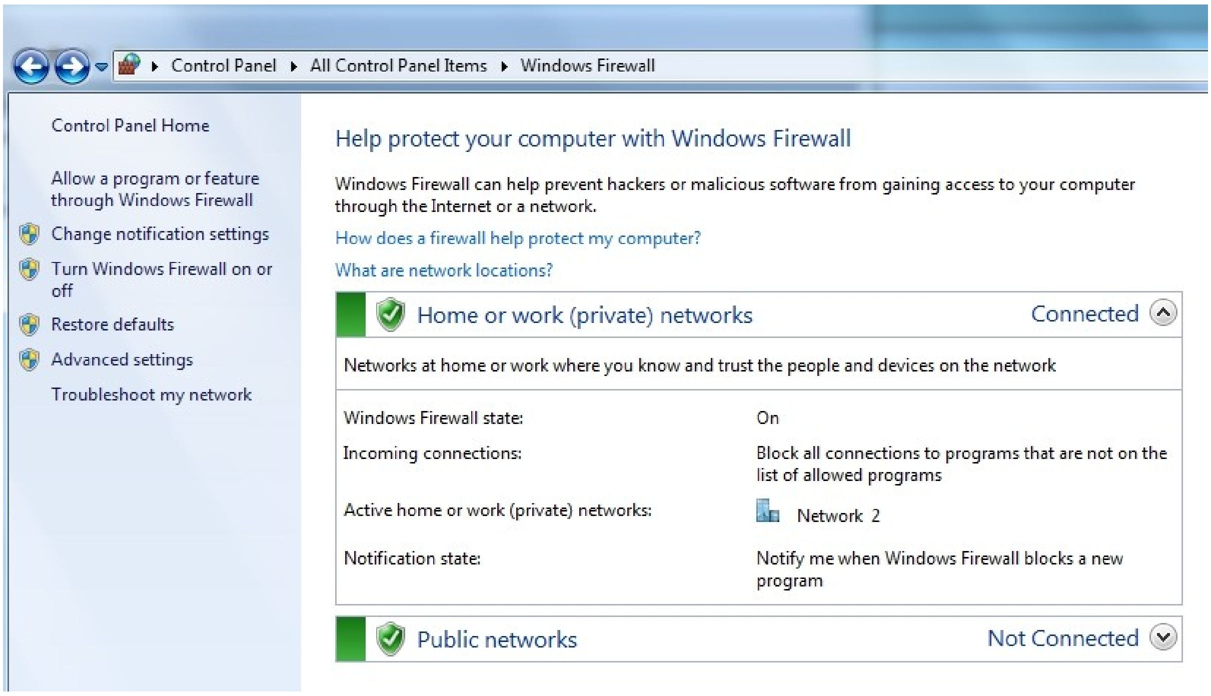
7.2.1 Setting up FileZilla FTP Server, Static public IP and Firewall

This is needed for GPRS/FTP only.



If you are having problems with setting up FileZilla Server to run behind Windows Firewall (specifically, it fails on "List" and the client receives a "Failed to receive directory listing" error), you must add the FileZilla Server application to Windows Firewall's Exceptions list. To do this, follow these steps:

1. Open Windows or other active Firewall under Control Panel.
2. If using Windows Vista, click "Change Settings"
3. Select the "Exceptions" tab.
4. Click "Add program..."
5. Do NOT select "FileZilla Server Interface" from the list, instead click on "Browse..."
6. Locate the directory you installed FileZilla Server to (normally "C:\Program Files\FileZilla Server\")
7. Double click or select "FileZilla server.exe" and press open (Once again, NOT "FileZilla Server Interface.exe")
8. Select "FileZilla server.exe" from the list and click "Ok"
9. Verify that "FileZilla server.exe" is added to the exceptions list and that it has a check mark in the box next to it
10. Press "Ok" to close the window
11. see more https://wiki.filezilla-project.org/Network_Configuration


Control Panel ▸ All Control Panel Items ▸ Windows Firewall
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network

Help protect your computer with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your computer through the Internet or a network.

How does a firewall help protect my computer?

What are network locations?

Home or work (private) networks — Connected

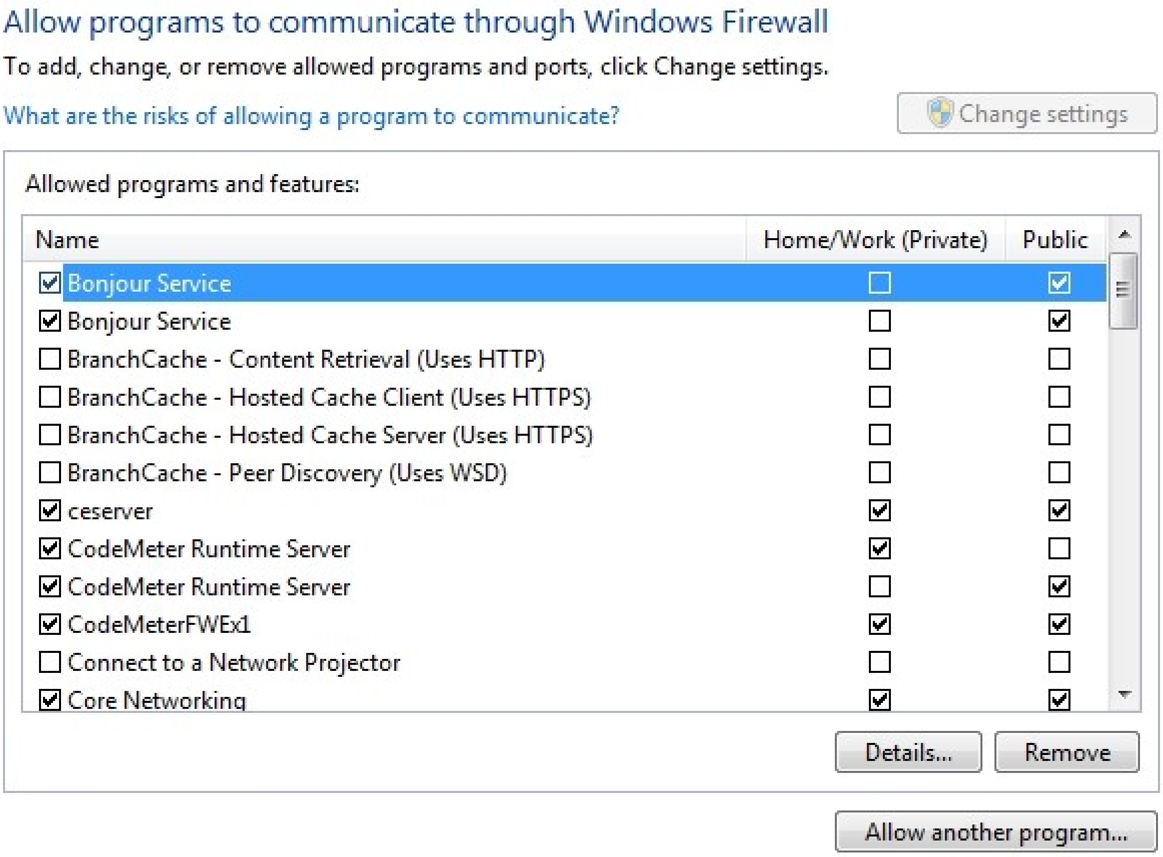
Networks at home or work where you know and trust the people and devices on the network

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the list of allowed programs
Active home or work (private) networks:	Network 2
Notification state:	Notify me when Windows Firewall blocks a new program

Public networks — Not Connected

Firewall settings in Windows 7 Profesional. Log in to windows with administrator user rights. Open control panel ->Firewall

Click Allow a program or feature…



Allow programs to communicate through Windows Firewall

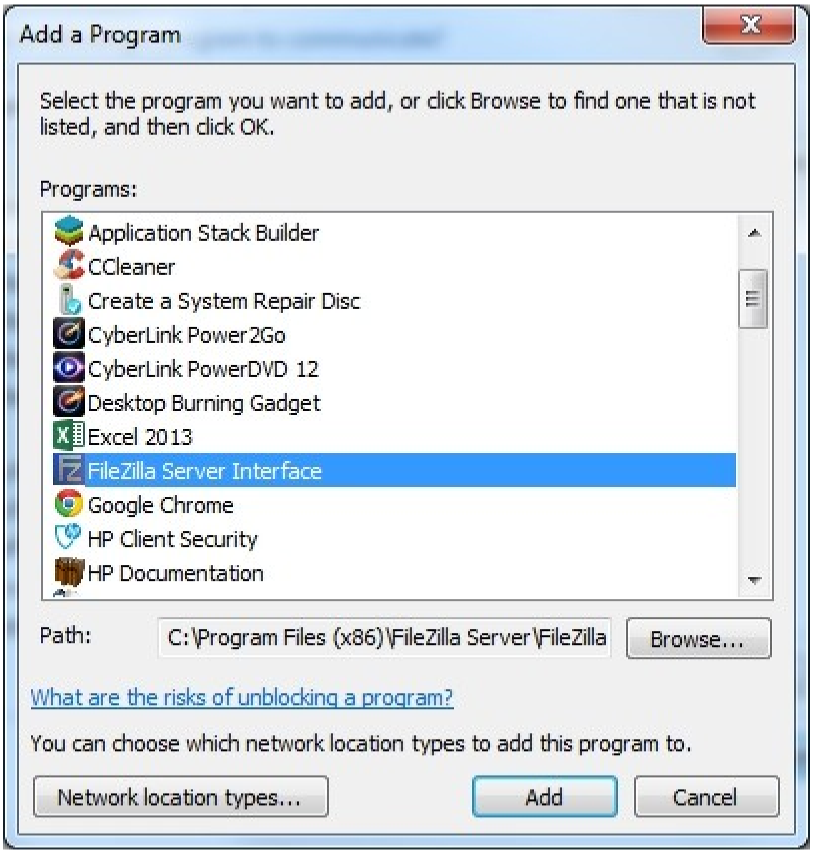
To add, change, or remove allowed programs and ports, click Change settings.

What are the risks of allowing a program to communicate?

Allowed programs and features:

Name	Home/Work (Private)	Public
☑ Bonjour Service	□	☑
☑ Bonjour Service	□	☑
□ BranchCache - Content Retrieval (Uses HTTP)	□	□
□ BranchCache - Hosted Cache Client (Uses HTTPS)	□	□
□ BranchCache - Hosted Cache Server (Uses HTTPS)	□	□
□ BranchCache - Peer Discovery (Uses WSD)	□	□
☑ ceserver	☑	☑
☑ CodeMeter Runtime Server	☑	□
☑ CodeMeter Runtime Server	□	☑
☑ CodeMeterFWEx1	☑	☑
□ Connect to a Network Projector	□	□
☑ Core Networking	☑	☑

Click Change settings and Allow another program…



Add a Program

Select the program you want to add, or click Browse to find one that is not listed, and then click OK.

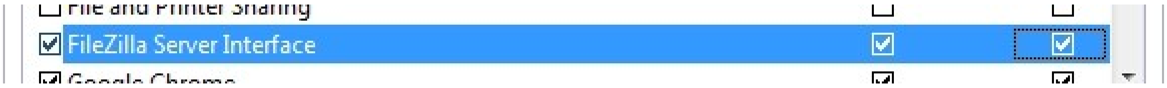
Programs:

- Application Stack Builder
- CCleaner
- Create a System Repair Disc
- CyberLink Power2Go
- CyberLink PowerDVD 12
- Desktop Burning Gadget
- Excel 2013
- FileZilla Server Interface
- Google Chrome
- HP Client Security
- HP Documentation

Path: C:\Program Files (x86)\FileZilla Server\FileZilla

You can choose which network location types to add this program to.

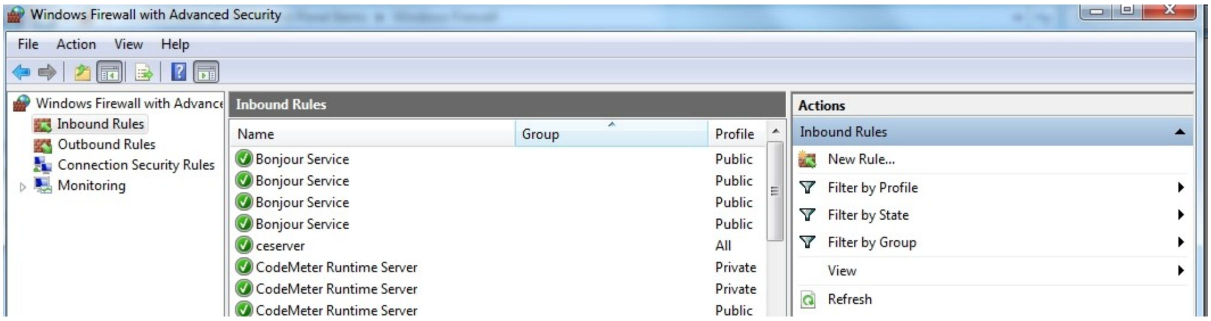
Select Filezilla server interface click add



☐ File and Printer Sharing	☐	☐
☑ FileZilla Server Interface	☑	☑
☑ Google Chrome	☑	☑

Set public network.

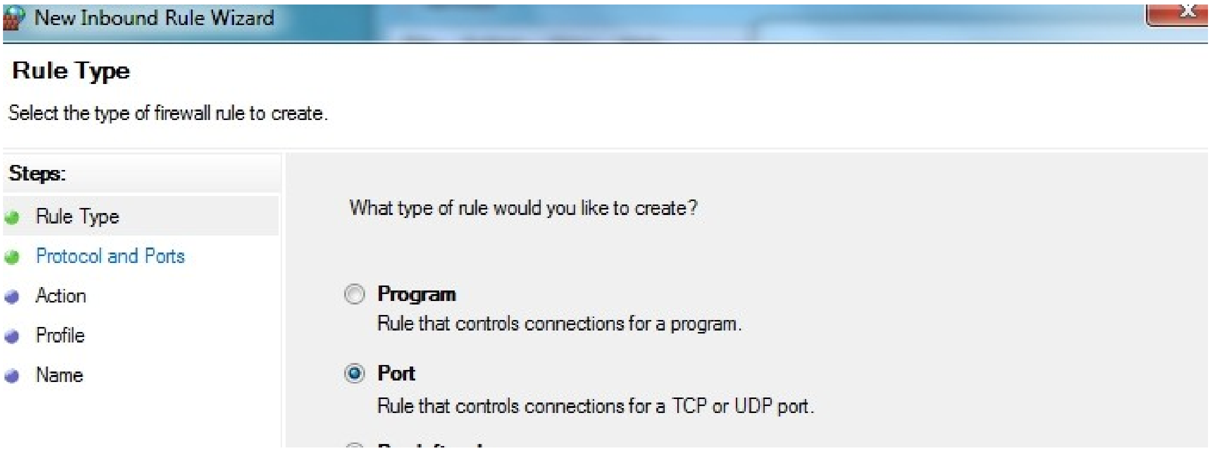
7.2.2 Open TCP port 21 for FTP



Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Inbound Rules

Name	Group	Profile
Bonjour Service		Public
Bonjour Service		Public
Bonjour Service		Public
Bonjour Service		Public
ceserver		All
CodeMeter Runtime Server		Private
CodeMeter Runtime Server		Private
CodeMeter Runtime Server		Public

Open also TCP port 21 in Firewall, which is used for FTP.



Rule Type

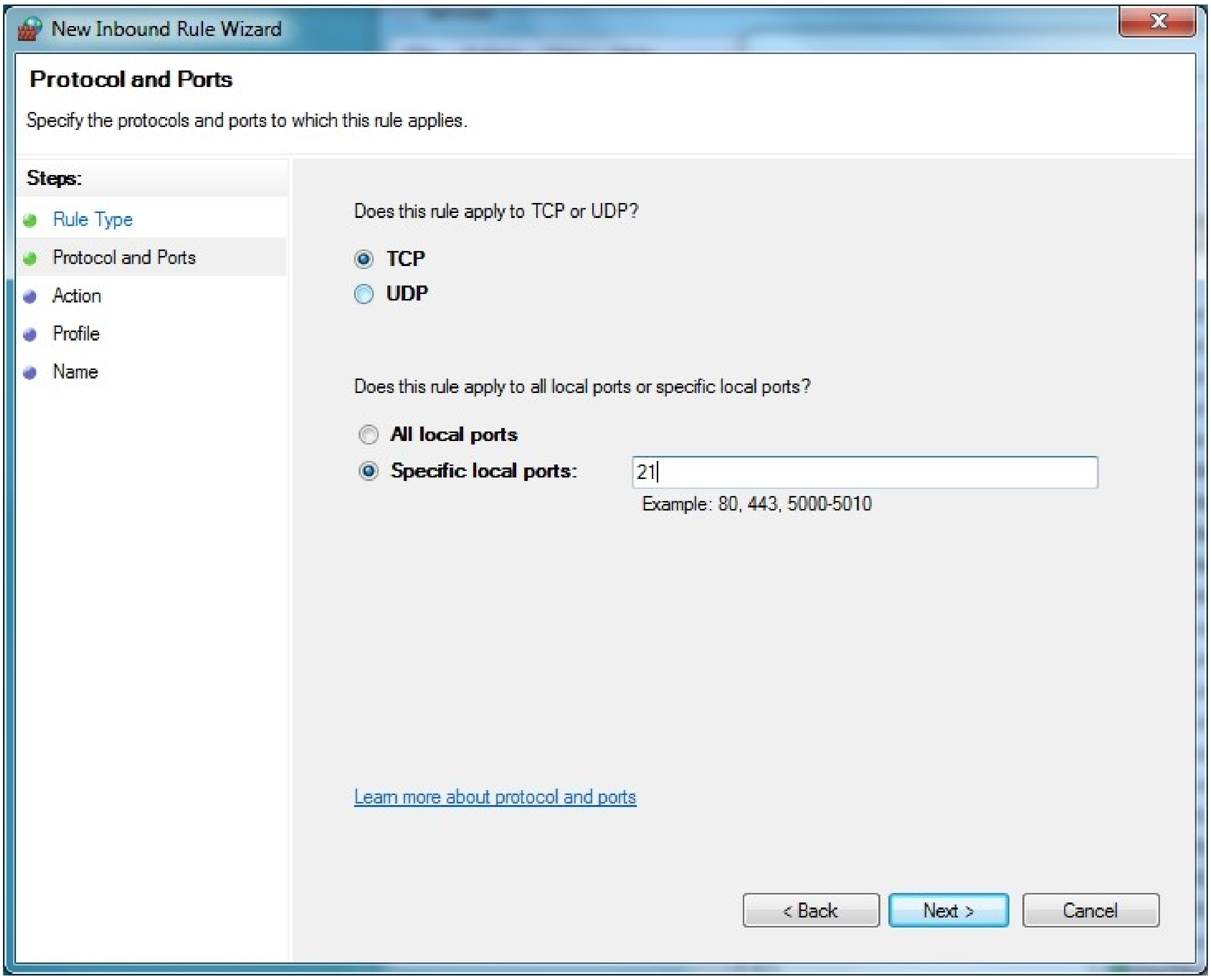
Select the type of firewall rule to create.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What type of rule would you like to create?

○ **Program**
Rule that controls connections for a program.

● **Port**
Rule that controls connections for a TCP or UDP port.



Protocol and Ports

Specify the protocols and ports to which this rule applies.

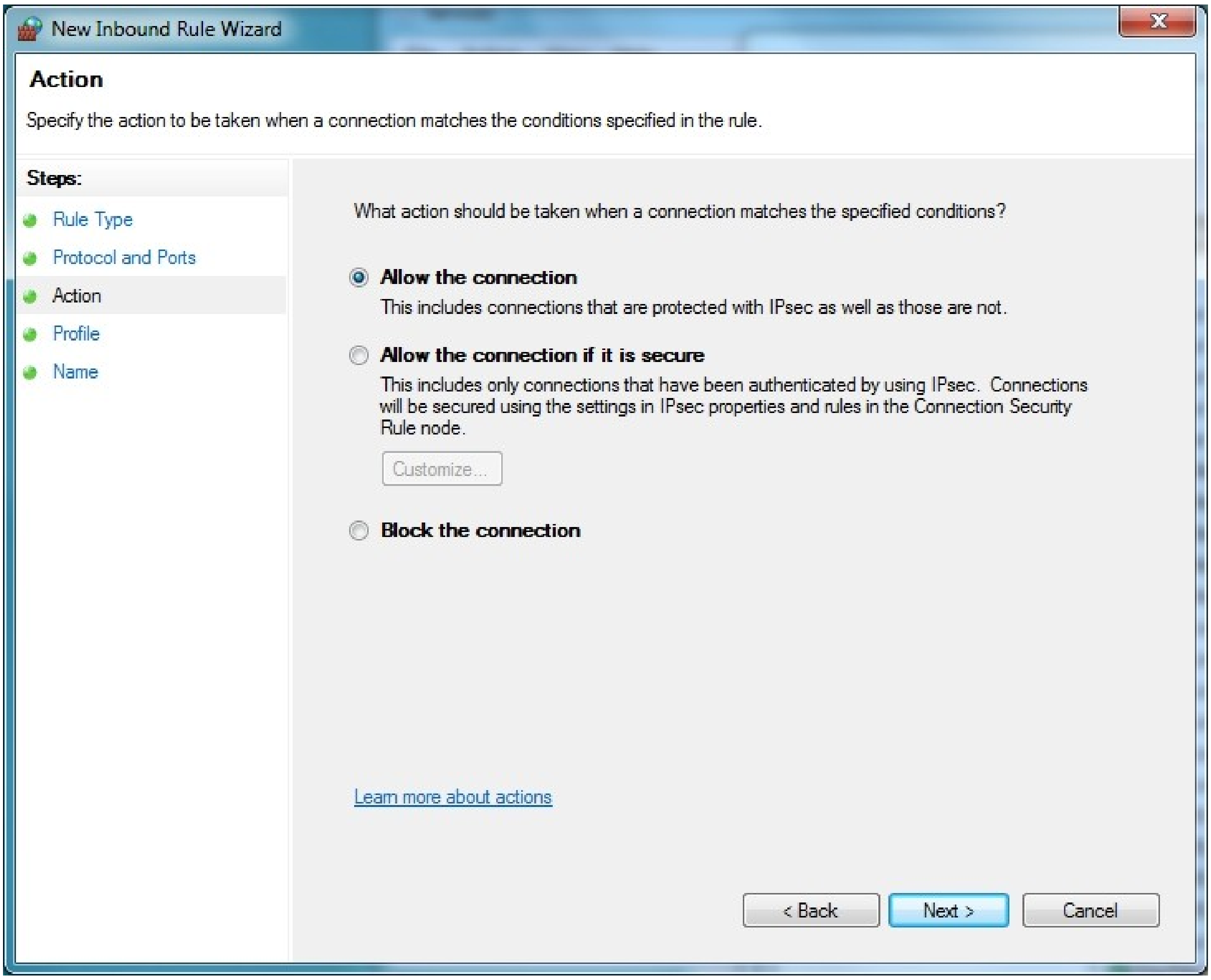
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

● **TCP**
○ **UDP**

Does this rule apply to all local ports or specific local ports?

○ **All local ports**
● **Specific local ports:** 21
Example: 80, 443, 5000-5010



New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:

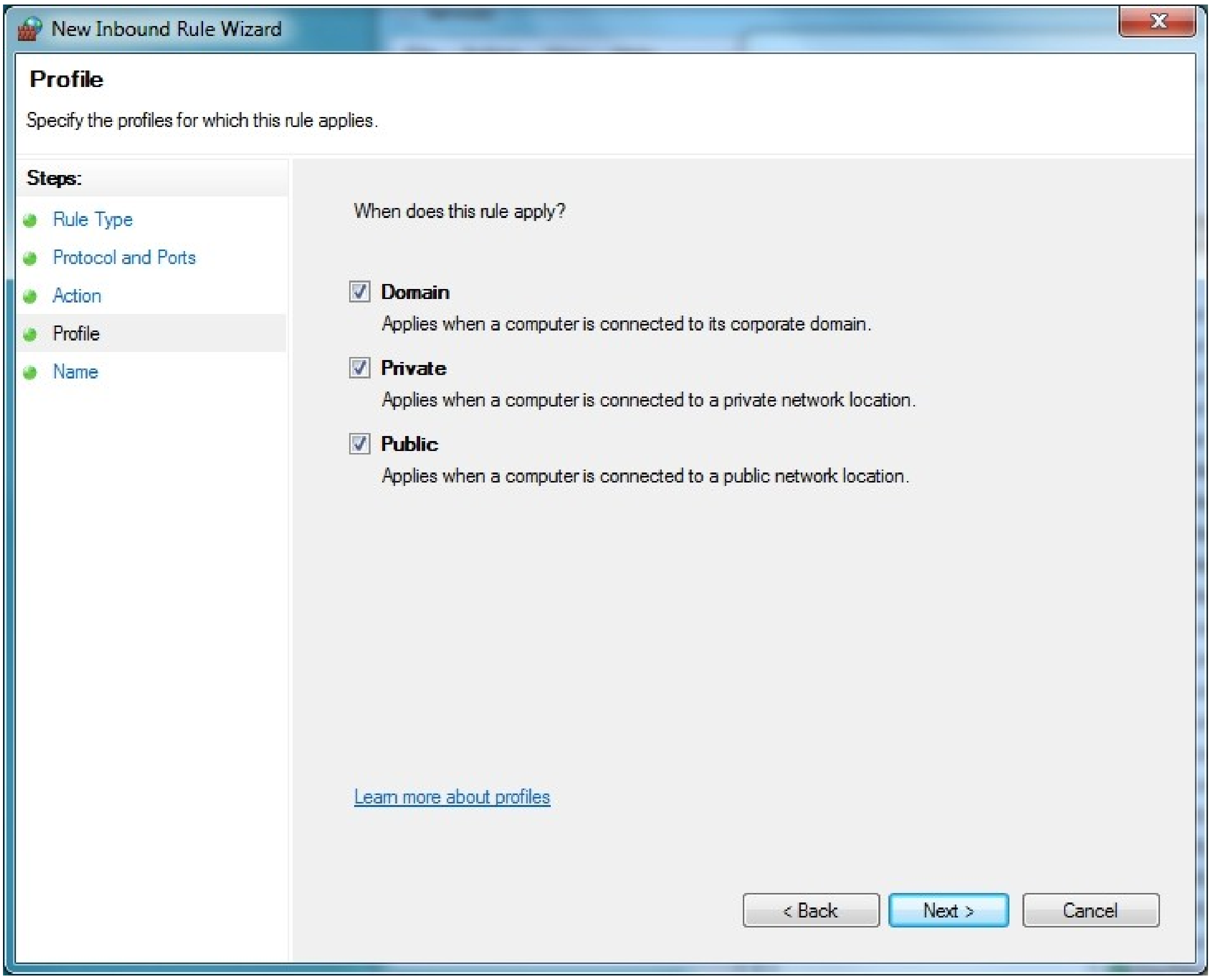
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

● **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

○ **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

○ **Block the connection**



New Inbound Rule Wizard

Profile

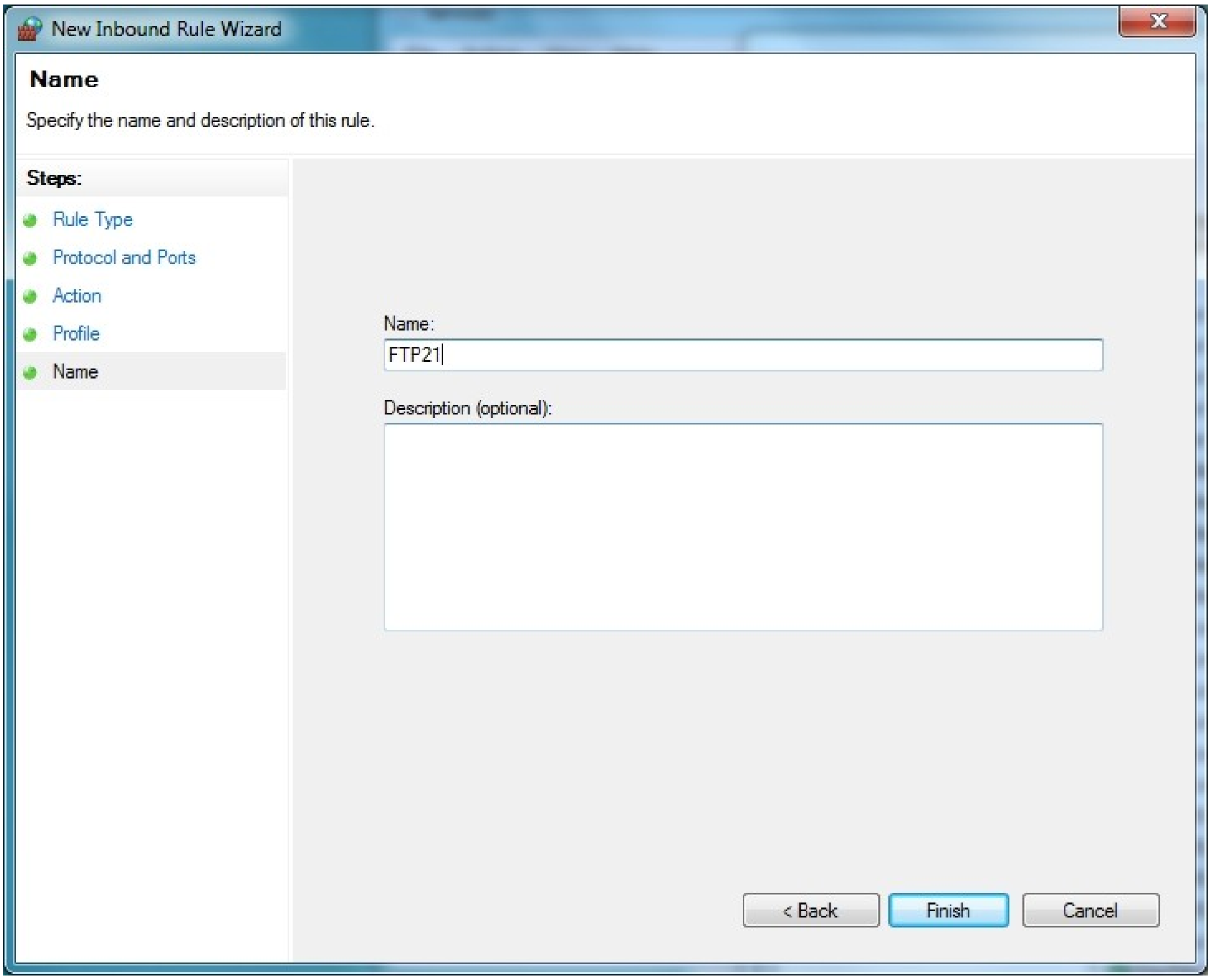
Specify the profiles for which this rule applies.

When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location.

☑ **Public**
Applies when a computer is connected to a public network location.



New Inbound Rule Wizard

Name

Specify the name and description of this rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

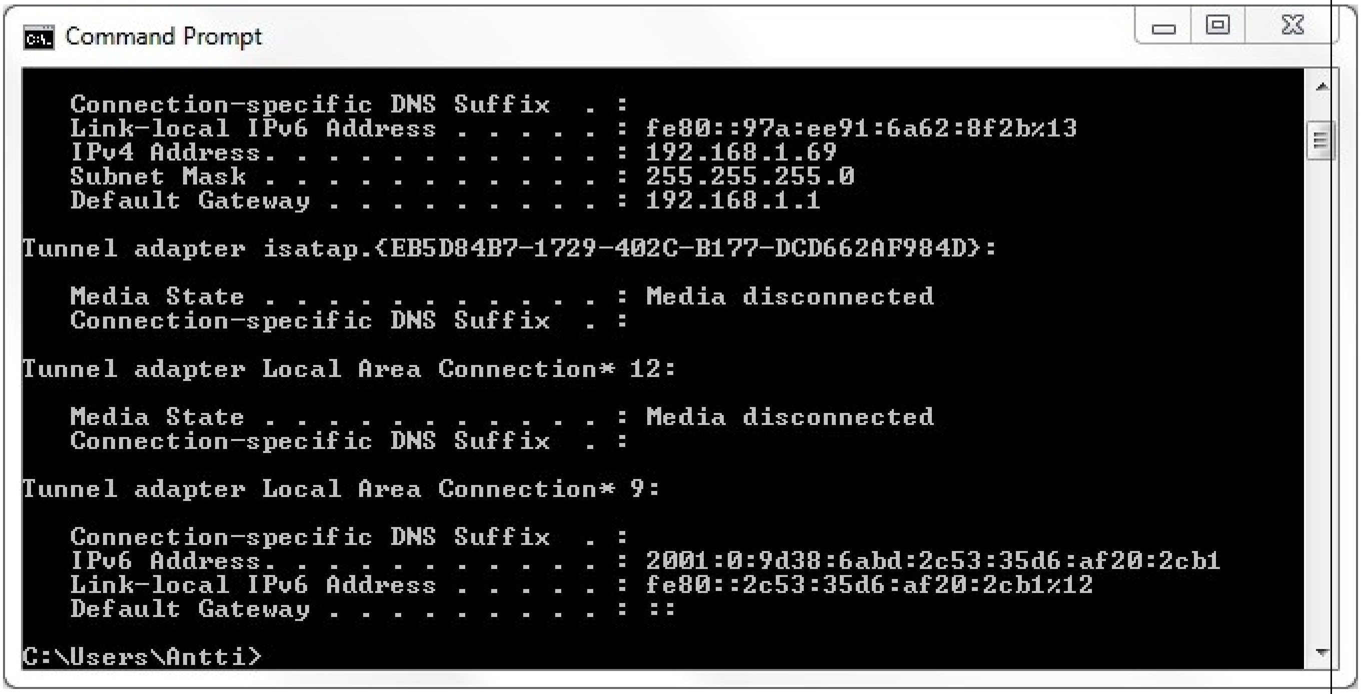
Name: FTP21

Description (optional):

Click Finish.

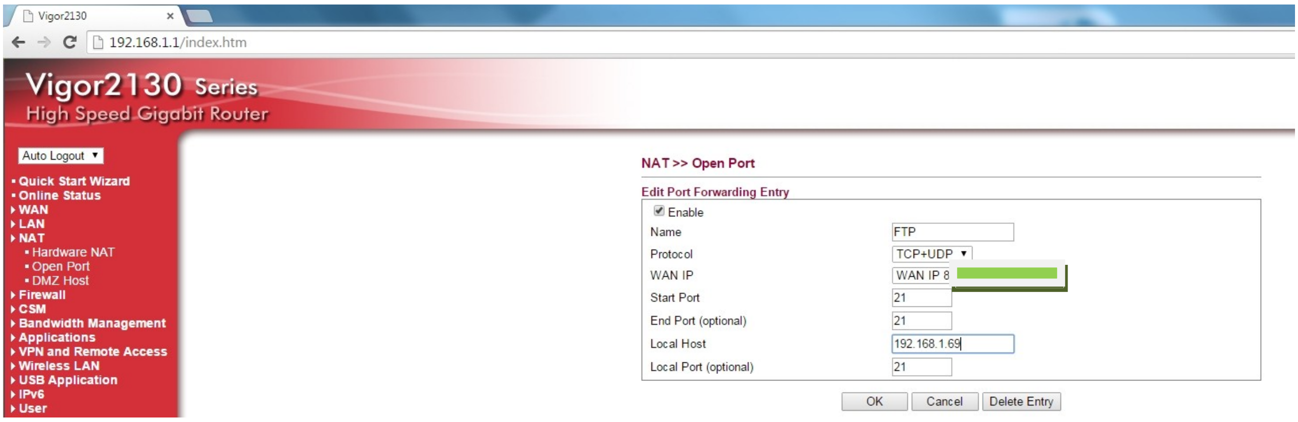
7.2.3 Forwarding TCP port 21 in router NAT setup

If your PC is in Intranet subnet and doesn't have public IP address, you might need to forward your public IP address's port 21 to your PC's Intranet IP address.
If your PC has public IP address, you don't need to do this.



```
   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::97a:ee91:6a62:8f2b%13
   IPv4 Address. . . . . . . . . . . : 192.168.1.69
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{EB5D84B7-1729-402C-B177-DCD662AF984D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 12:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 9:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:9d38:6abd:2c53:35d6:af20:2cb1
   Link-local IPv6 Address . . . . . : fe80::2c53:35d6:af20:2cb1%12
   Default Gateway . . . . . . . . . : ::

C:\Users\Antti>
```

Check your PC's IPv4 address. Start->All Programs->Accessories->Cmd Promt



NAT >> Open Port

Edit Port Forwarding Entry

☑ Enable

Name: FTP

Protocol: TCP+UDP

WAN IP: WAN IP 8 [redacted]

Start Port: 21

End Port (optional): 21

Local Host: 192.168.1.69

Local Port (optional): 21

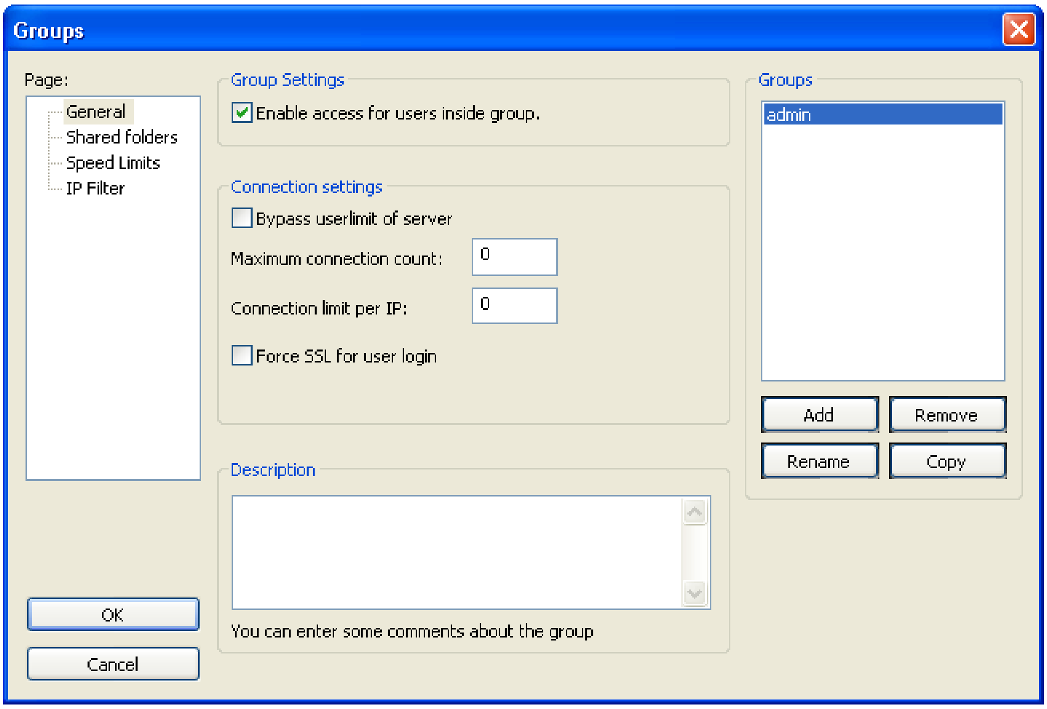
Open router configuration, normally using IE http://192.168.1.1 search for NAT setup. The above picture is just an example.

Make the configuration. Do not copy the IP addresses - Use your own WAN IP and local host IP addresses.

7.2.4 Order Static public IP address

Note! You will (might) need static public IP address for your SCADA server PC, otherwise your application will not work long time. Dymanic IP address might change when you reboot your computer. Please consult your Internet network operator and order public static IP address for your network connection. For testing you can have dynamic IP address. To check your IP address you can open http://www.whatismyip.com/

7.2.5 Setting up Filezilla FTP server software:



Groups

Page:
- General
- Shared folders
- Speed Limits
- IP Filter

Group Settings

☑ Enable access for users inside group.

Connection settings

□ Bypass userlimit of server

Maximum connection count: 0

Connection limit per IP: 0

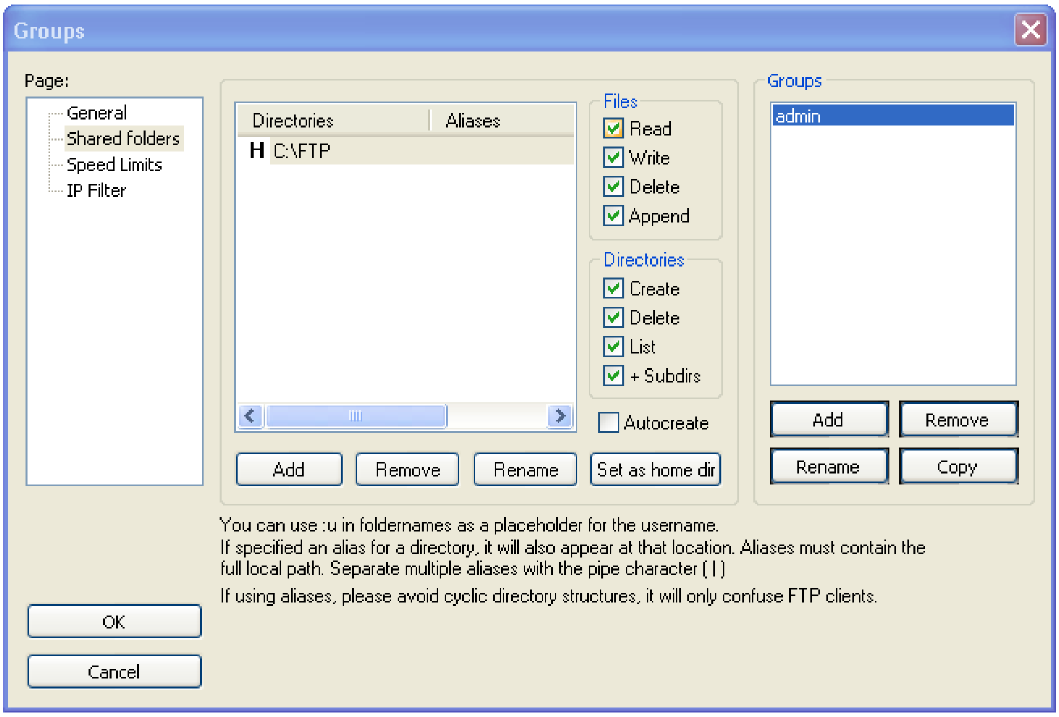
□ Force SSL for user login

Description

You can enter some comments about the group

Groups: admin

Edit->Groups
Add (user group) admin
OK



Directories	Aliases
H C:\FTP	

Files
☑ Read
☑ Write
☑ Delete
☑ Append

Directories
☑ Create
☑ Delete
☑ List
☑ + Subdirs

□ Autocreate

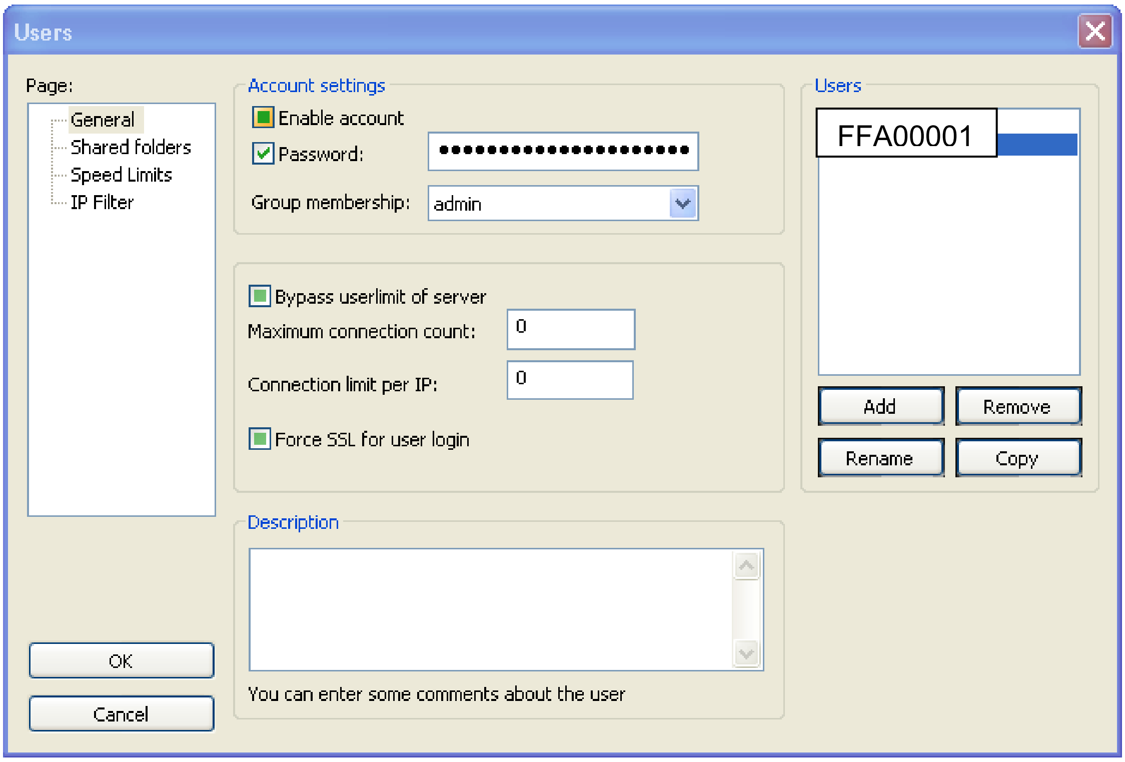
Groups: admin

You can use :u in foldernames as a placeholder for the username.
If specified an alias for a directory, it will also appear at that location. Aliases must contain the full local path. Separate multiple aliases with the pipe character (|)
If using aliases, please avoid cyclic directory structures, it will only confuse FTP clients.

Click edit->Groups->Shared folders
(Create directory C:\FTP using windows explorer)

Add home directory (C:\FTP) and full permissions, click set as home dir
Click OK to save settings!

CONFIGURATIONS



Users

Page:
- General
- Shared folders
- Speed Limits
- IP Filter

Account settings

☑ Enable account

☑ Password: ●●●●●●●●●●●●●●●●●●●●●

Group membership: admin

☑ Bypass userlimit of server

Maximum connection count: 0

Connection limit per IP: 0

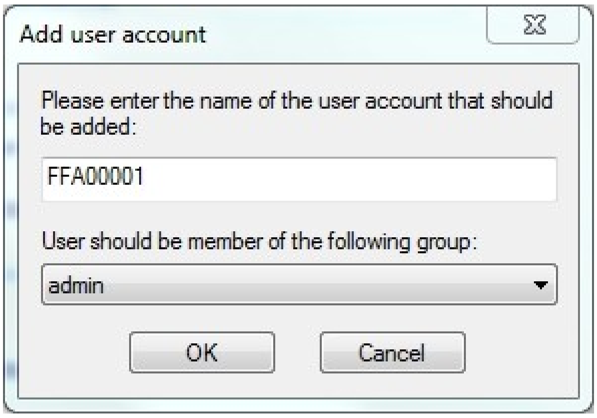
☑ Force SSL for user login

Description

You can enter some comments about the user

Users: FFA00001

Select edit->Users



Add user account

Please enter the name of the user account that should be added:

FFA00001

User should be member of the following group:

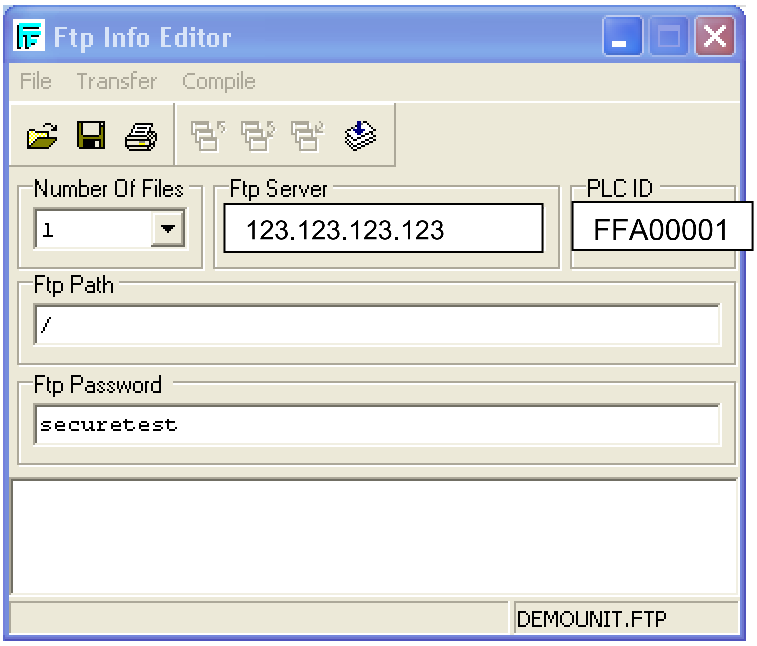
admin

add user name.

<u>If you are using AutoLog GSM-PLCs you should add one user for every GSM PLC</u>, user name should be same as PLC ID which is defined from the AutoLog GsmProgrammer software. e.g. (FFA00001).

(PLC_ID can be set using GSMProgrammer. View-> FTP info)

CONFIGURATIONS



Number Of Files: 1

Ftp Server: 123.123.123.123

PLC ID: FFA00001

Ftp Path: /

Ftp Password: securetest

DEMOUNIT.FTP

PLC_ID should have **8 HEX characters!**

(HEX characters are 0123456789ABCDEF)

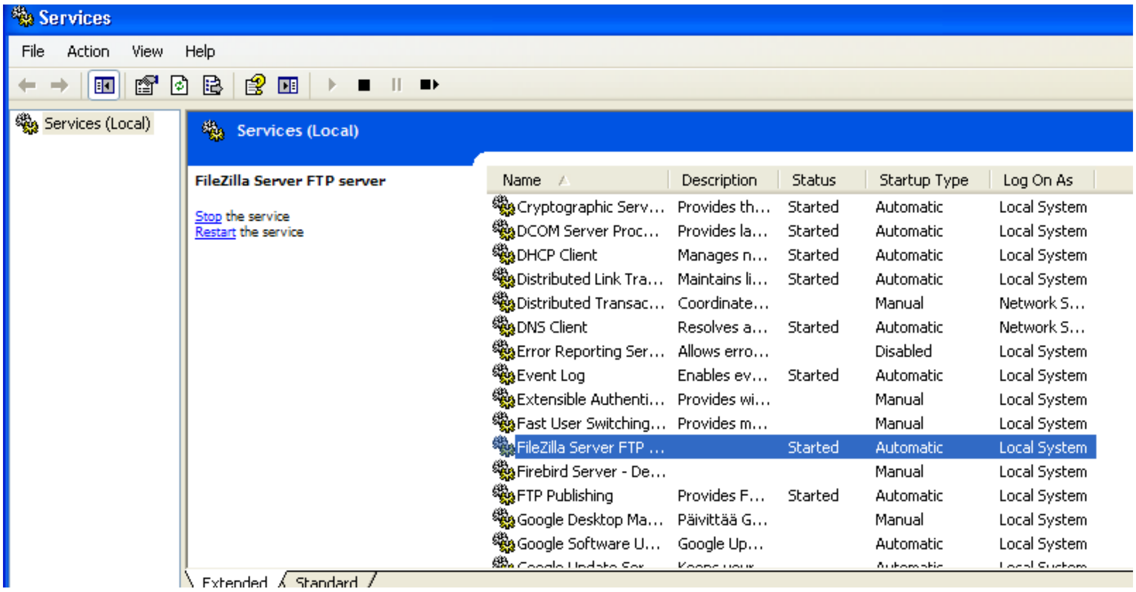
(After changing the PLC ID, you need to compile the project and transfer it to GSM-PLC and start the PLC, Look GSM-PLC User Manual how to do that!
Every GSM-PLC you have, should have unique PLC_ID, you can use numbers 0-9 and letters A-F.

Ftp path “/” means that the files are send to the root directory (=home directory) of the FTP server.

Ftp password should be same as defined in GsmProgrammer Ftp password: e.g. “securetest”.

Click OK.

7.2.6 Restarting the FTP Server



FileZilla Server FTP server

Stop the service
Restart the service

Name	Description	Status	Startup Type	Log On As
Cryptographic Serv…	Provides th…	Started	Automatic	Local System
DCOM Server Proc…	Provides la…	Started	Automatic	Local System
DHCP Client	Manages n…	Started	Automatic	Local System
Distributed Link Tra…	Maintains li…	Started	Automatic	Local System
Distributed Transac…	Coordinate…		Manual	Network S…
DNS Client	Resolves a…	Started	Automatic	Network S…
Error Reporting Ser…	Allows erro…		Disabled	Local System
Event Log	Enables ev…	Started	Automatic	Local System
Extensible Authenti…	Provides wi…		Manual	Local System
Fast User Switching…	Provides m…		Manual	Local System
FileZilla Server FTP …		Started	Automatic	Local System
Firebird Server - De…			Manual	Local System
FTP Publishing	Provides F…	Started	Automatic	Local System
Google Desktop Ma…	Päivittää G…		Manual	Local System
Google Software U…	Google Up…		Automatic	Local System
Google Update Ser…	Keeps your…		Automatic	Local System

You might need to restart the server after this change!

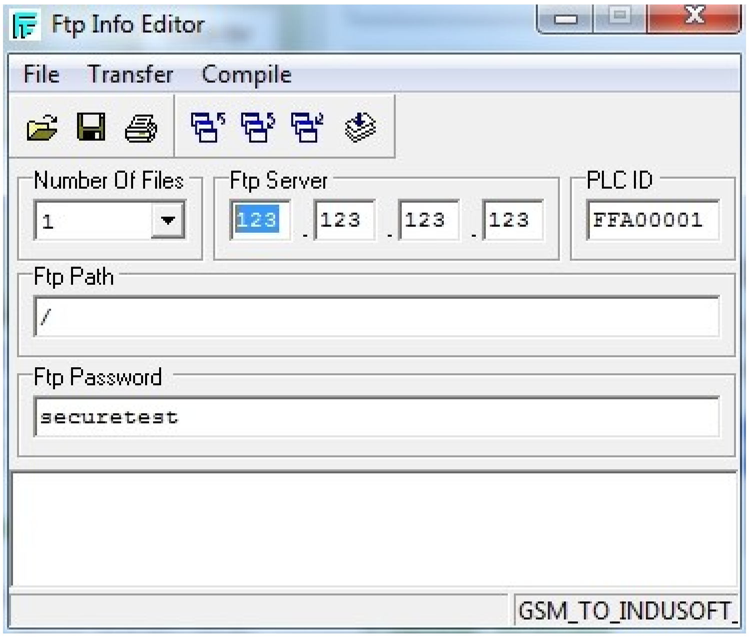
You can check the status of the Filezilla server in Control Panel->Administrative Tools->Services

Click restart.

7.2.7 Setting up the GSM-PLC to send FTP files over GPRS

Open GSMProgrammer

View->FTP info



Number Of Files: 1

Ftp Server: 123 . 123 . 123 . 123

PLC ID: FFA00001

Ftp Path: /

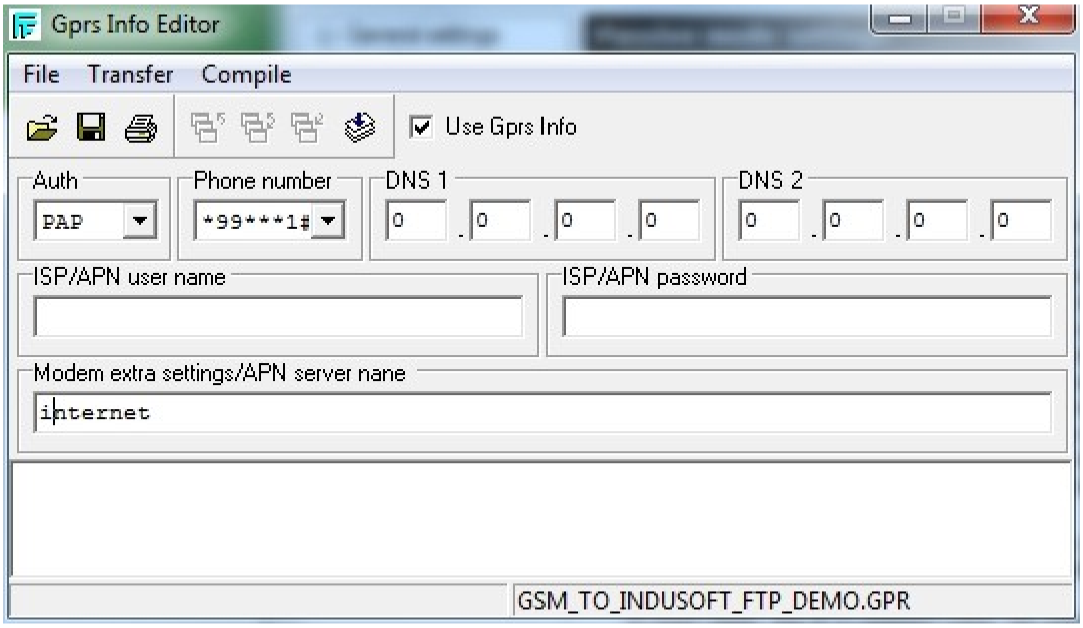
Ftp Password: securetest

GSM_TO_INDUSOFT

Check your public IP address. www.whatismyipaddress.com

Write your public IP address to the Ftp server fields.
Click save

View->Gprs info



☑ Use Gprs Info

Auth: PAP

Phone number: *99***1#

DNS 1: 0 . 0 . 0 . 0

DNS 2: 0 . 0 . 0 . 0

ISP/APN user name:

ISP/APN password:

Modem extra settings/APN server nane: internet

GSM_TO_INDUSOFT_FTP_DEMO.GPR

Check your mobile operators (SIM card supplier's) Internet APN name.
You can google this using words like "[your operator name] internet access point name".
Check also if you need to use some DNS IP addresses and what is the authentication method.

The above setting work in Finland, DNA mobile operator's network.

Compile->Compile

Transfer ->Transfer Project

Click Green RUN button.

Set GSM-PLC's DIP 4 = OFF, insert SIM card (PIN code request is turned off), and reboot the GSM-PLC (activates GSM/GPRS modem).

For GSM-20 put the GSM modem cable back to GSM-PLC's serial port 1.

Check that the FTP_Parser –program is stopped. If not right click on this icon on the system tray and stop it.

If your GSMProgrammer FTP settings, Firewall settings, SIM card and GSM modem settings and FTP server settings are working, you should receive files to your FTP server's home directory.

1 file / 60 seconds

Note! If FTP_parser is running it will read the FTP files from the home directory and move the old files to Old_files directory so in this case you might not see FTP files in this directory.

Check the system tray if the FTP_parser is running. Close it to verify that the FTP files are coming to the server's home directory.

7.2.8 If you experience problems with FTP file receiving try the following:

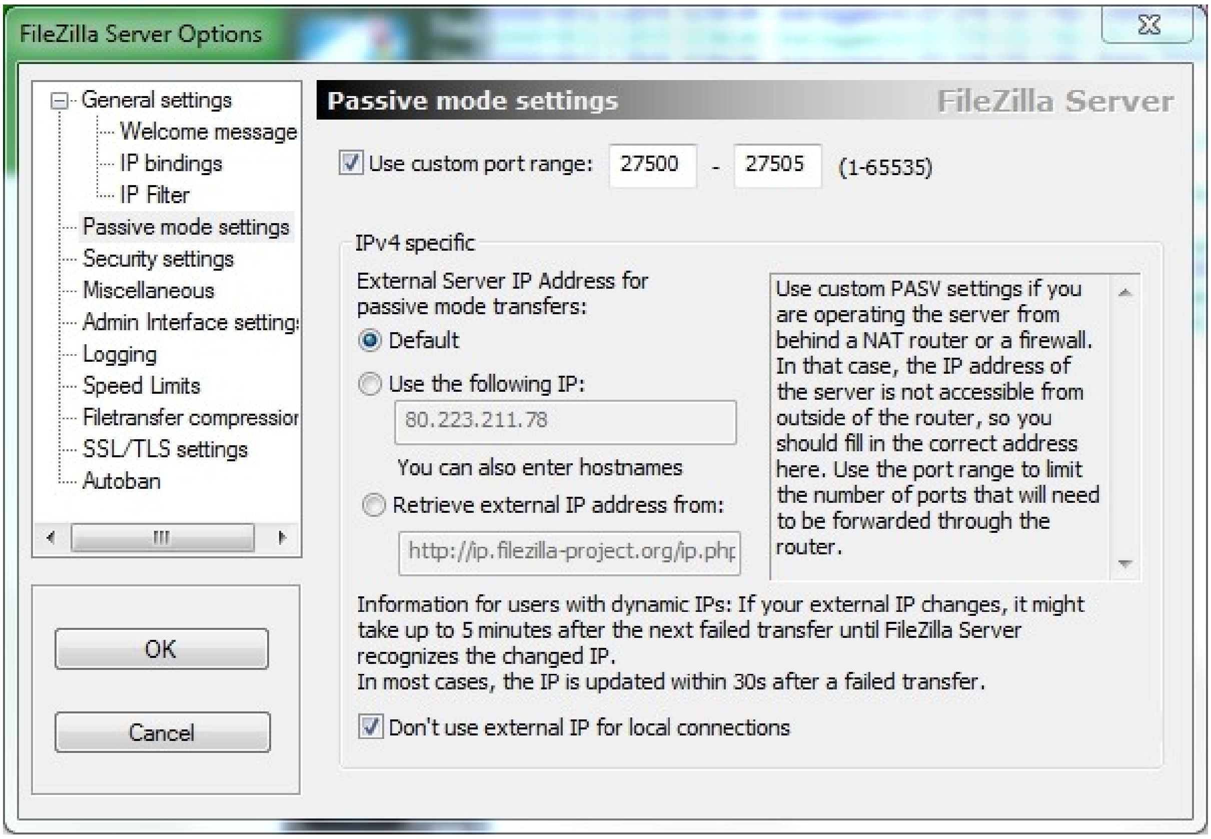
1. Disable firewall temporarily and test if it works!
2. After you got it working turn on the firewall again and make some firewall exception rules and configure Filezille settings according to the following instruction:

https://wiki.filezilla-project.org/Network_Configuration#Firewalls

e.g. this helped when the NAT port forwarding is used:

Open also ports e.g. 27500-27510 (or define the range freely >2000)

Open the Filezilla->Edit->Settings->Passive mode settings



☑ Use custom port range: 27500 - 27505 (1-65535)

IPv4 specific

External Server IP Address for passive mode transfers:

● Default

○ Use the following IP: 80.223.211.78

You can also enter hostnames

○ Retrieve external IP address from: http://ip.filezilla-project.org/ip.php

Use custom PASV settings if you are operating the server from behind a NAT router or a firewall. In that case, the IP address of the server is not accessible from outside of the router, so you should fill in the correct address here. Use the port range to limit the number of ports that will need to be forwarded through the router.

Information for users with dynamic IPs: If your external IP changes, it might take up to 5 minutes after the next failed transfer until FileZilla Server recognizes the changed IP.
In most cases, the IP is updated within 30s after a failed transfer.

☑ Don't use external IP for local connections

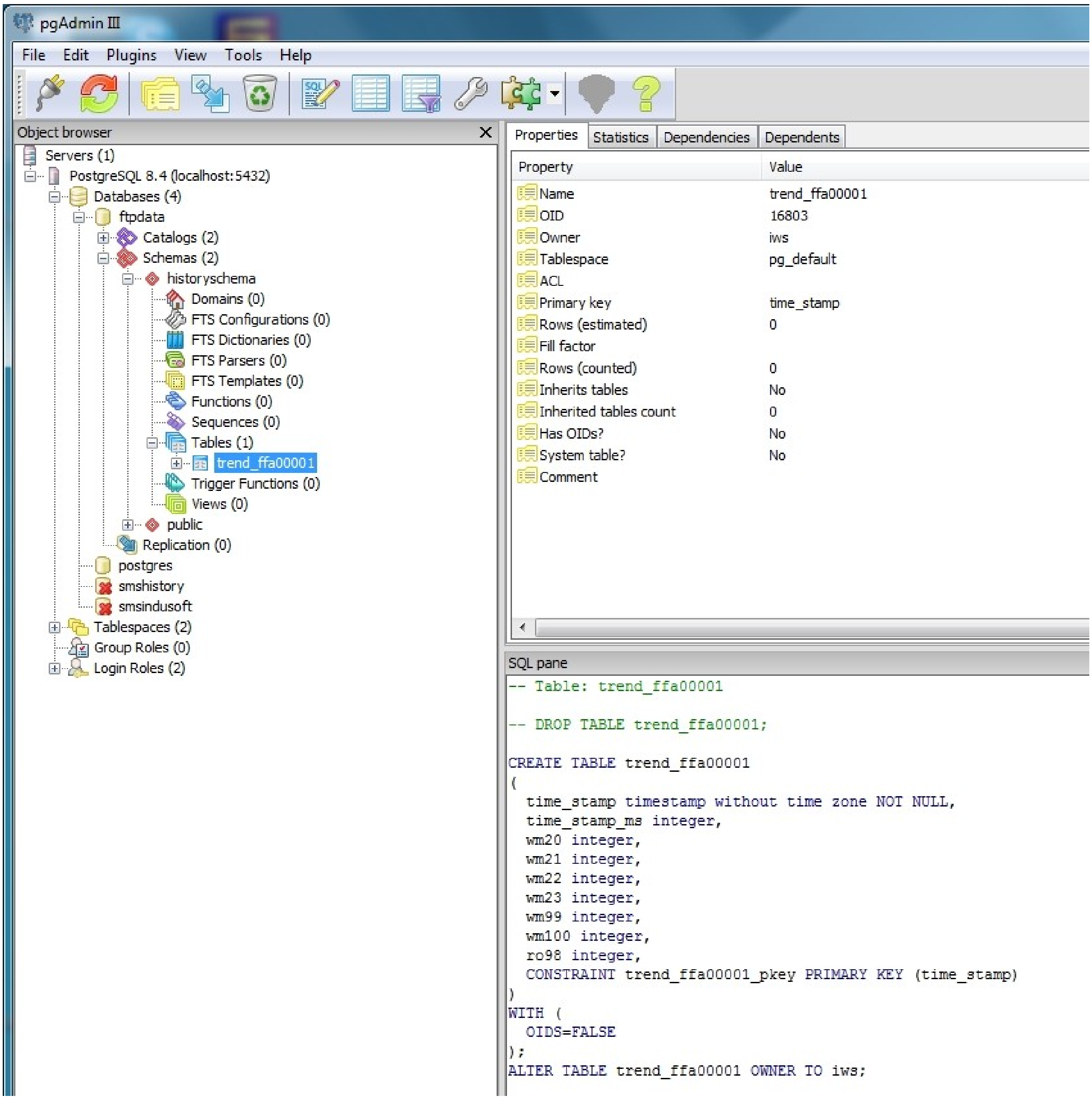
After you have received files to your FTP directory you can move to next step.

7.3 Testing the FTP_parser with GSM-PLC

Test table was created earlier in this manual.

TABLE trend_ffa00001

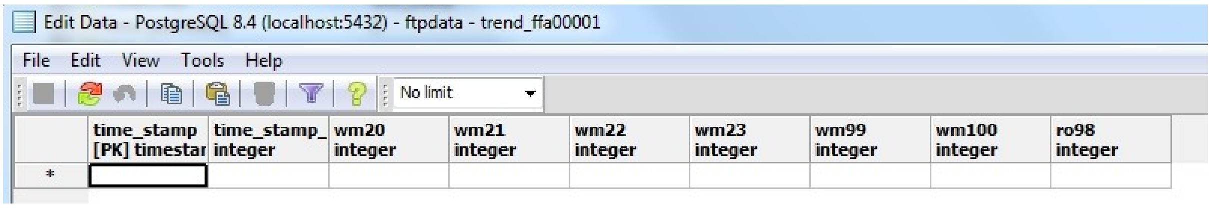
Start->All Programs->PostGreSQL->pgAdminIII



Property	Value
Name	trend_ffa00001
OID	16803
Owner	iws
Tablespace	pg_default
ACL	
Primary key	time_stamp
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Has OIDs?	No
System table?	No
Comment	

```
-- Table: trend_ffa00001

-- DROP TABLE trend_ffa00001;

CREATE TABLE trend_ffa00001
(
  time_stamp timestamp without time zone NOT NULL,
  time_stamp_ms integer,
  wm20 integer,
  wm21 integer,
  wm22 integer,
  wm23 integer,
  wm99 integer,
  wm100 integer,
  ro98 integer,
  CONSTRAINT trend_ffa00001_pkey PRIMARY KEY (time_stamp)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE trend_ffa00001 OWNER TO iws;
```

Browse to the ftp database and right click trend_ffa00001 table.


Edit Data - PostgreSQL 8.4 (localhost:5432) - ftpdata - trend_ffa00001
File Edit View Tools Help
No limit

	time_stamp [PK] timestar	time_stamp_ integer	wm20 integer	wm21 integer	wm22 integer	wm23 integer	wm99 integer	wm100 integer	ro98 integer
*									

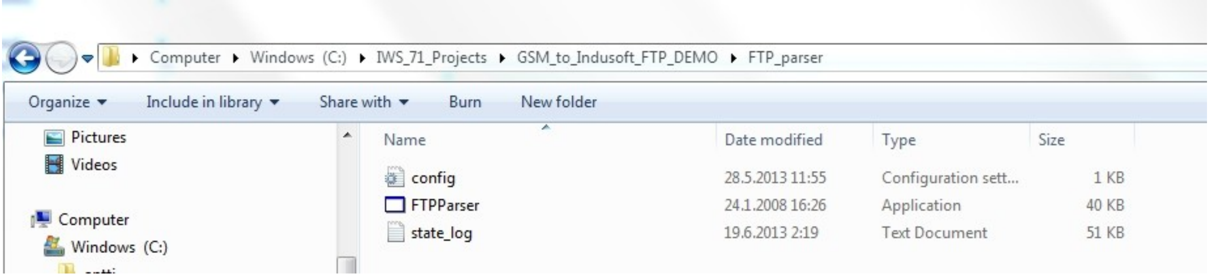
View data->View all rows
The table is still empty

TESTING FTP DEMO

Open the windows explorer and locate your GSM_to_Indusoft_FTP_demo (Indusoft project directory).

You can copy the “GSM_to_Indusoft_FTP_demo” project directory under the following folder:

C:\IWS_71_Projects\



Name	Date modified	Type	Size
config	28.5.2013 11:55	Configuration sett...	1 KB
FTPParser	24.1.2008 16:26	Application	40 KB
state_log	19.6.2013 2:19	Text Document	51 KB

FTP_parser driver is in the FTP_parser subdirectory. Open it.

Open config.ini and check the configuration.

Check that the DataDir=’C:\FTP’ (or same as your FTP server’s home directory)

Check that database is ftpdata
Uid=iws
Pwd=ffauto (or your own password which you have defined when creating iws user to the postGreSQL database.

You can check also the state.log which should be empty.

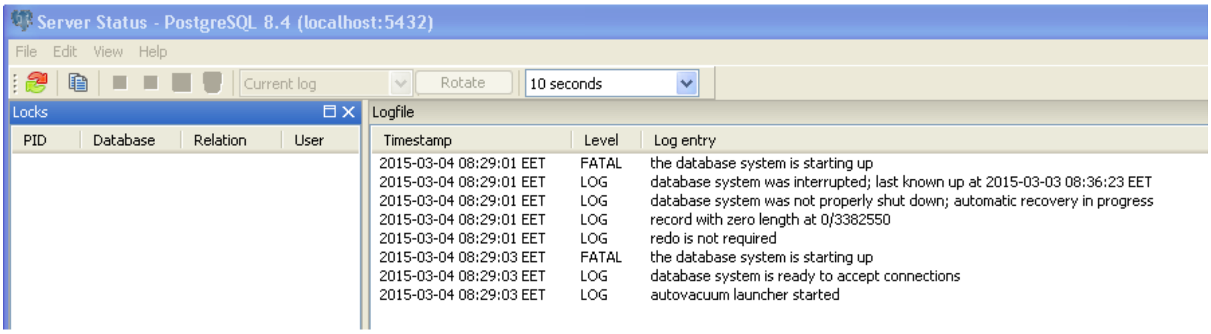
Youi can open the C:\FTP directory to see the ftp files. These files will be parsed to the PostGreSQL database with the ftp_parser –driver.

Now you can start the ftp_parser by double –clicking the “FTPParser –application file”.

If you test setup is working the table should be filled with data coming from the GSM-PLC. Click table refresh button.

If you cannot see data after about 2 minutes, one thing you can check is the PosGreSQL’s server status which is useful for solving database related problems.

Tools->Server Status



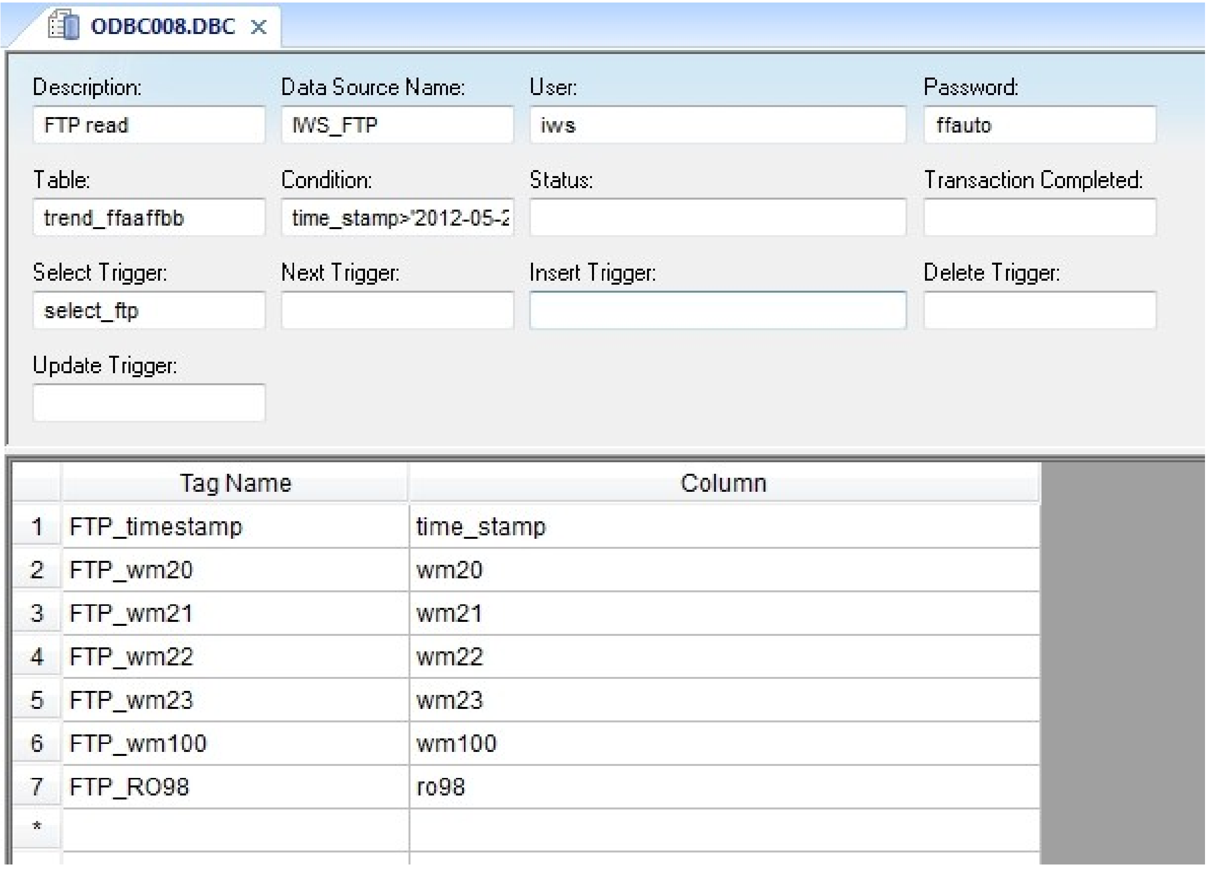
```
Timestamp                Level   Log entry
2015-03-04 08:29:01 EET  FATAL   the database system is starting up
2015-03-04 08:29:01 EET  LOG     database system was interrupted; last known up at 2015-03-03 08:36:23 EET
2015-03-04 08:29:01 EET  LOG     database system was not properly shut down; automatic recovery in progress
2015-03-04 08:29:01 EET  LOG     record with zero length at 0/3382550
2015-03-04 08:29:01 EET  LOG     redo is not required
2015-03-04 08:29:03 EET  FATAL   the database system is starting up
2015-03-04 08:29:03 EET  LOG     database system is ready to accept connections
2015-03-04 08:29:03 EET  LOG     autovacuum launcher started
```

TESTING FTP DEMO

Check the LogFile window for tracking problems.

7.4 Reading FTP data to Indusoft Web Studio

Open Indusoft Web Sudio->ODBC Task sheet



ODBC008.DBC

Description: FTP read	Data Source Name: IWS_FTP	User: iws	Password: ffauto
Table: trend_ffaaffbb	Condition: time_stamp>'2012-05-2	Status:	Transaction Completed:
Select Trigger: select_ftp	Next Trigger:	Insert Trigger:	Delete Trigger:
Update Trigger:			

	Tag Name	Column
1	FTP_timestamp	time_stamp
2	FTP_wm20	wm20
3	FTP_wm21	wm21
4	FTP_wm22	wm22
5	FTP_wm23	wm23
6	FTP_wm100	wm100
7	FTP_RO98	ro98
*		

Make the above configurations.

Make new Boolean type tag for “select_ftp”.

Use the following condition to sort the data in the table when select trigger is used so that the latest value will be read to the tag names from the Table’s colums. The time_stamp> clause will just check that the saved time_stamp has correct value and filter others.

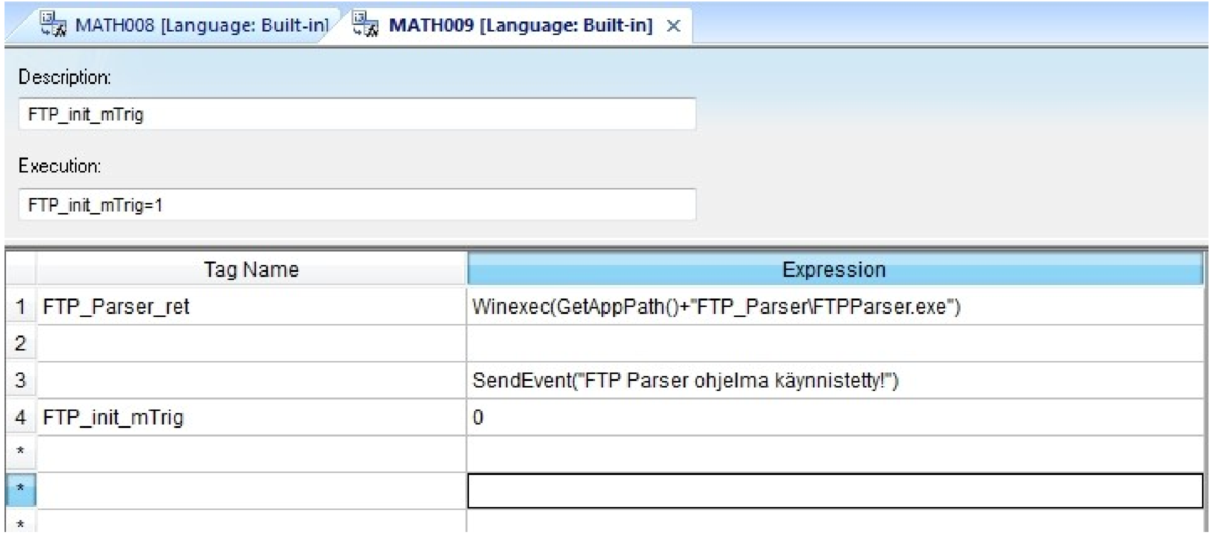
time_stamp>'2012-05-28 10:56:00' ORDER BY time_stamp desc

FTP_timestamp	time_stamp
FTP_wm20	wm20
FTP_wm21	wm21
FTP_wm22	wm22
FTP_wm23	wm23
FTP_wm100	wm100
FTP_RO98	ro98

You can copy the above values to the Tag Name, Column -fields. You need to create new tags for every tag name.

FTP_timestamp (type string)
FTP_wmXX (type integer)
FTP_RO98 (type integer

7.4.1 Starting the FTP_parser from the Indusoft Web Studio Startup:



Description:

FTP_init_mTrig

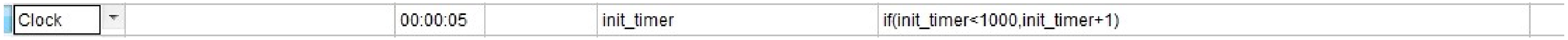
Execution:

FTP_init_mTrig=1

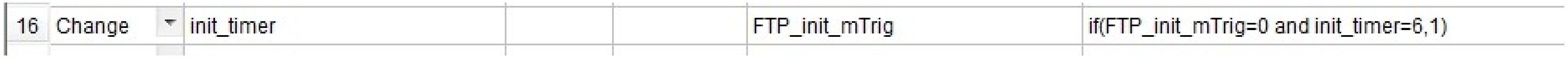
	Tag Name	Expression
1	FTP_Parser_ret	Winexec(GetAppPath()+"FTP_Parser\FTPParser.exe")
2		
3		SendEvent("FTP Parser ohjelma käynnistetty!")
4	FTP_init_mTrig	0
*		
*		
*		

You can use the following Math task to start FTP_parser automatically.

Make the following scheduler task –lines to start the FTP_init_mTrig 30 seconds after starting the application:

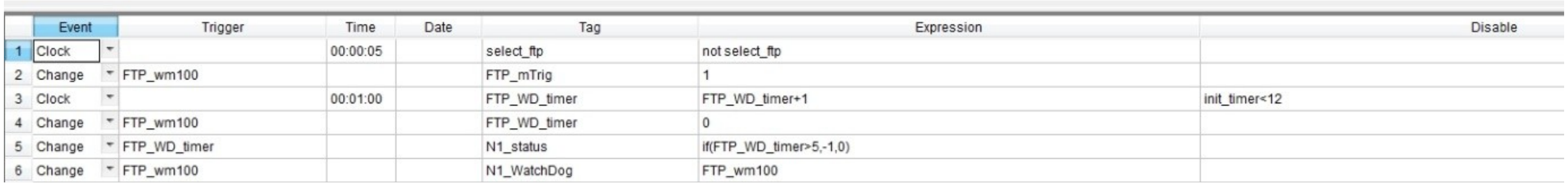


Clock		00:00:05		init_timer	if(init_timer<1000,init_timer+1)



16	Change	init_timer			FTP_init_mTrig	if(FTP_init_mTrig=0 and init_timer=6,1)

7.4.2 Controlling ODBC task from the scheduler –task to read data from the database



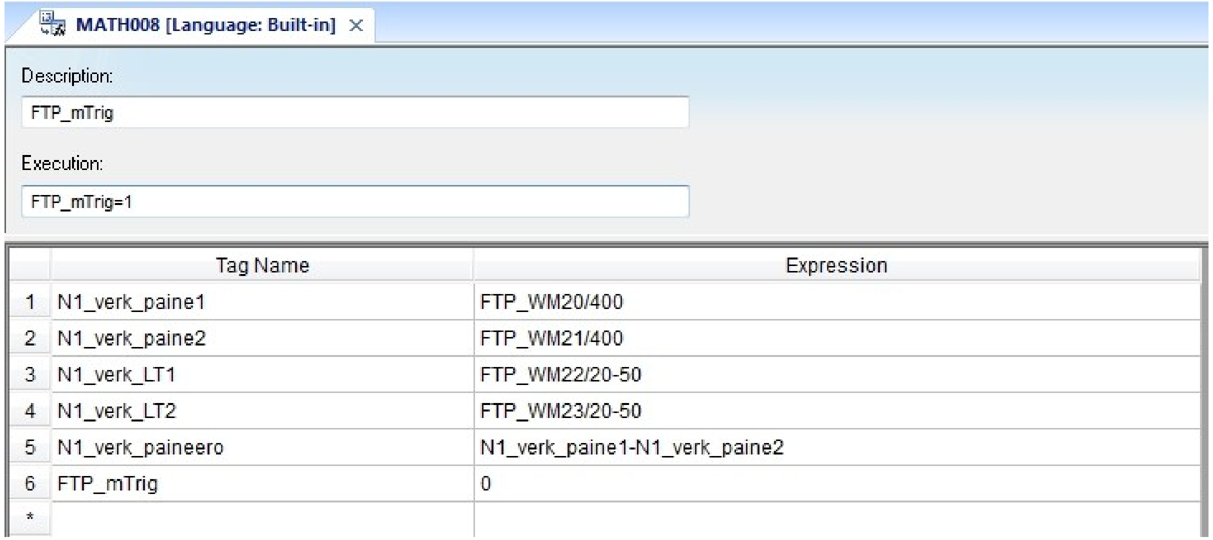
	Event	Trigger	Time	Date	Tag	Expression	Disable
1	Clock		00:00:05		select_ftp	not select_ftp	
2	Change	FTP_wm100			FTP_mTrig	1	
3	Clock		00:01:00		FTP_WD_timer	FTP_WD_timer+1	init_timer<12
4	Change	FTP_wm100			FTP_WD_timer	0	
5	Change	FTP_WD_timer			N1_status	if(FTP_WD_timer>5,-1,0)	
6	Change	FTP_wm100			N1_WatchDog	FTP_wm100	

Clock	00:00:05	select_ftp	not select_ftp	
Change	FTP_wm100	FTP_mTrig	1	
Clock	00:01:00	FTP_WD_timer	FTP_WD_timer+1	init_timer<12
Change	FTP_wm100	FTP_WD_timer	0	
Change	FTP_WD_timer	N1_status	if(FTP_WD_timer>5,-1,0)	
Change	FTP_wm100	N1_WatchDogFTP_wm100		

FTP_mTrig = Starts the Math –task which scales the raw analog values to engineering units (See the next page!)

WD = WatchDog which is used to check that the FTP communication is working.

7.4.3 Math –task which scales the raw analog values to engineering units



Description:
FTP_mTrig

Execution:
FTP_mTrig=1

	Tag Name	Expression
1	N1_verk_paine1	FTP_WM20/400
2	N1_verk_paine2	FTP_WM21/400
3	N1_verk_LT1	FTP_WM22/20-50
4	N1_verk_LT2	FTP_WM23/20-50
5	N1_verk_paineero	N1_verk_paine1-N1_verk_paine2
6	FTP_mTrig	0
*		

Paine = pressure (in Finnish)
LT = LämpöTila = temperature (in Finnish)

See complete Indusoft web studio FTP demo application to learn more!

7.5 Configuring GsmProgrammer and Pump demo monitoring –template application

This demo application is used to demonstrating how the DB2SMS driver and Indusoft Web Studio is used and configured to monitor incoming SMS measurement data, alarms etc.

Look GSM-PLC service manual for basic instructions!

Transfer the Pump_Demo_application to the GSM-PLC.

See the source code in Appendix!

8 PART 3 – Using DB2SMS with Indusoft Web Studio

This part of the manual describes how to use DB2SMS with Indusoft Web Studio (IWS) SCADA. DB2SMS –program is GSM/SMS –driver, which allows Web Studio to communicate with AutoLog GSM-RTUs bi-directionally. This manual should be used with IWS Demo project, which is made with IWS v.6.1. The purpose of this document with DEMO project is to show all essential features of using AutoLog GSM-RTUs with Web Studio so system designers can get as fast getting started process as possible.

PostgreSQL's smsindusoft - database is used just as an interface and not for permanent storing of measurement data. Measurement data is read to Indusoft Web Studio and stored to another database e.g. PostGreSQL's smshistory –database using IWS's ODBC task. Also other database types can be used, but it needs some modifications in ODBC task sheets.

FF-Automation would appreciate to get any feedback of this manual and DEMO project.

Any questions and feedback can be sent to antti.moijanen@ff-automation.com or info@ff-automation.com

8.1 Required software components.

- Indusoft Webstudio 6.1 + SP2 or later
- DB2SMS 2.1 or later (supplied with DEMO project and manual)
  - .Net Framework 2.0 (Not needed for Windows 7 or if it is already installed) (http://www.microsoft.com/downloads/details.aspx?familyid=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en)
  - GSM-modem (you can order this from FF-Automation)
- PostgreSQL database
  - PostgreSQL (http://www.postgresql.org/ftp/binary/v8.2.3/win32/)

8.2 MyTest (short description of DEMO IWS project)

8.2.1 User Interface

MyTest DEMO –project's user interface includes versatile features for settings configuration values in the interface -database.

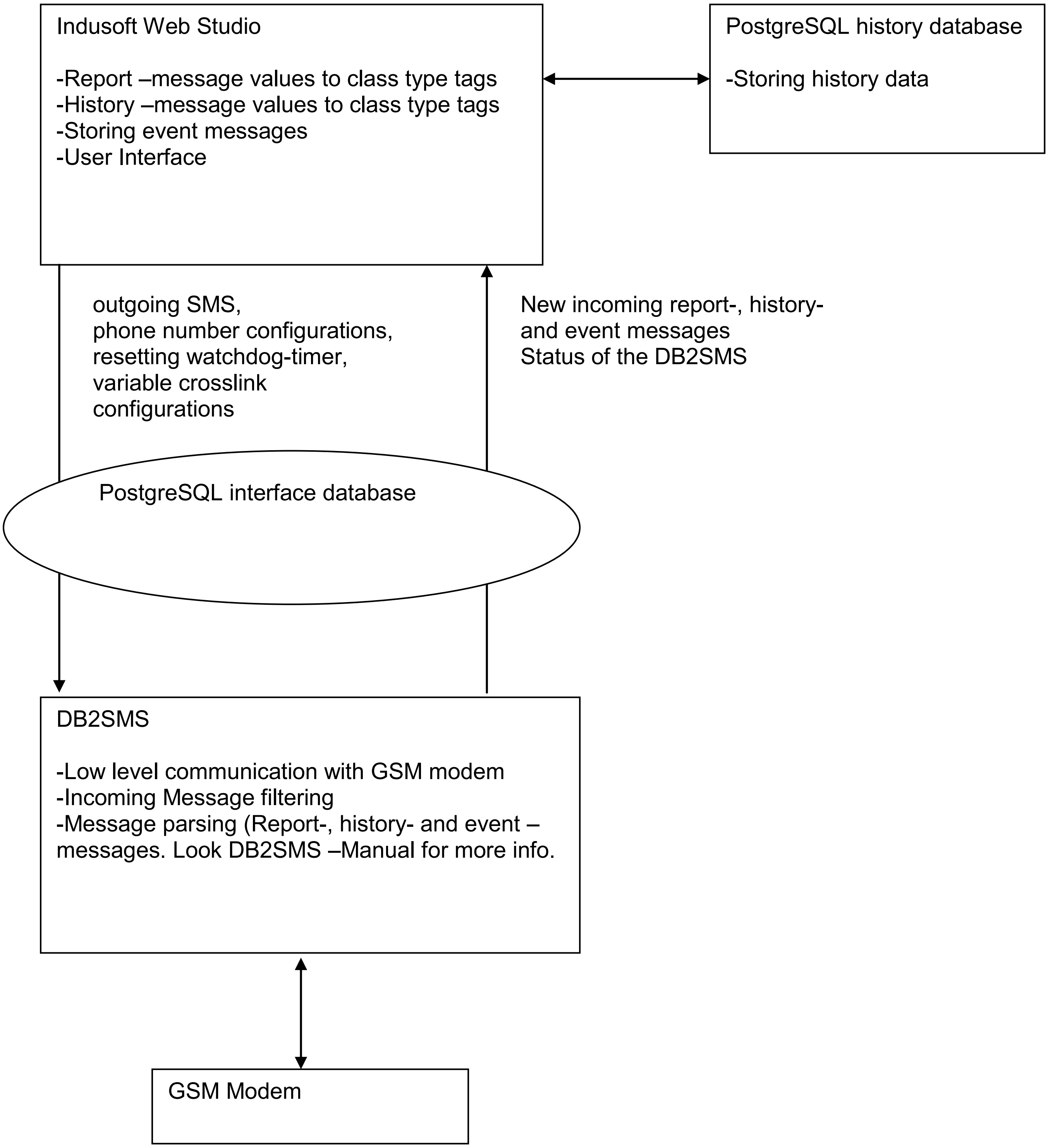
There is a ready-made trend object, that can be used to show the measurement values from the measurement history database.

User interface demonstrates also how to send SMS.

8.2.2 Temperature Alarm

MyTest application is sending alarm message to all GSM numbers in the Phonebook_Sending table if the PLC[1].Temperature value goes above 40°C. PLC is configured to send Report message every hour. PLC[x].Temperature is scaled from the PLC[x].AI1 variable (0…4096 to -50...150) using scheduler task. Alarm is created using Alarm –task.

8.3 Diagram of program functonality


Indusoft Web Studio
-Report –message values to class type tags
-History –message values to class type tags
-Storing event messages
-User Interface
PostgreSQL history database
-Storing history data
outgoing SMS,
phone number configurations,
resetting watchdog-timer,
variable crosslink
configurations
New incoming report-, history-
and event messages
Status of the DB2SMS
PostgreSQL interface database
DB2SMS
-Low level communication with GSM modem
-Incoming Message filtering
-Message parsing (Report-, history- and event –
messages. Look DB2SMS –Manual for more info.
GSM Modem

9 Using MyTest -User Interface

9.1 Phone numbers

The interface database has two phone number lists:
Phonebook_Sending –table includes numbers where the Group SMS messages are send.
Phonebook_Receiving –table includes numbers which are allowed to send incoming messages.

DB2SMS copies the value from Phonebook_Receiving | Name –field to Received_ * | Name –field.

* = Report, History or Events

Only those phone numbers are active in phonebook_* tables, which Enabled –field is <>0.

9.2 Sending SMS –messages from IWS

Sending SMS –message from the user interface is quite simple. User just writes the values to flag and message tags and triggers the saving tag.

9.2.1 Sending group message

1. Write DB_SendAll.Flag = 1
2. Write to tag DB_SendAll.Message the wanted SMS message
3. Trigger the interface database ODBC write procedure. Changing the value of DBC_SendAll.Insert does this. (DBC_SendAll.Insert = NOT DBC_SendAll.Insert)

Group SMS is sent to all active GSM numbers in the Phonebook_Sending –table.

9.2.2 Sending message to defined number

1. Write DB_SendSingle.Flag = 1
2. Write to tag DB_SendSingle.Phone the wanted phone number
3. Write to tag DB_SendSingle.Message the wanted message
4. Trigger the interface database ODBC write procedure. This is done by changing the value of DBC_SendSingle.Insert
   (DBC_SendSingle.Insert = NOT DBC_SendSingle.Insert)

9.3 Reading the error status (DB2SMS program)

You must trigger the DBC_ErrorMessage.Select –tag to read the status of the DB2SMS –program. The Error status –message can be read from the DB_ErrorMessage -tag. This is done automatically in every

9.4 Receiving SMS -messages

9.4.1 Event -messages

When IWS detects that new Event –messages have arrived to the interface database (Received_Events | Flag = 1) it just generates IWS's SendEvent –function. SendEvent – function stores the event to the event database and it can be seen using the Event object. If the event –message starts with word "ALARM" it will be send using group SMS sending to all phone numbers that are configured in Phonebook_sending –table.

9.4.2 History -messages

When IWS detects that new History –messages have arrived to the interface database (Received_History | Flag = 1) it writes the value to the history database. IWS selects the history table name using values from sender's *Name* (Received_History | Name) and *Variable* (Received_History | Variable) fields. Table name is format trend_{Name}_{variable} e.g. trend_PLC_AI0.

ODBC –task is not creating tables to the history database so user has to create those manually. See 5.4.4

9.4.3 Report -messages

When IWS detects that new Report –messages have arrived to the interface database (Received_report | Flag = 1) it stores the value (Received_report | Value) to class variable named Received_report | Name.Received_report | Variable. E.g if sender's name is PLC and variable's name is WM0, then value is stored to class variable PLC.WM0. Note that PLC is defined in IWS to be a class variable that has many members, one of which is WM0.

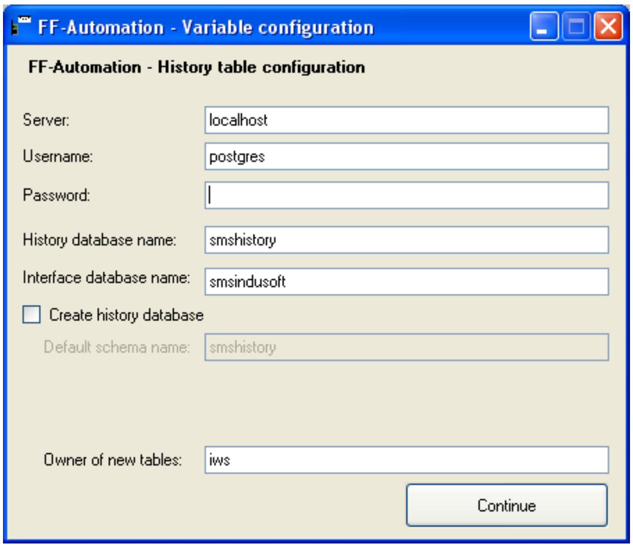
Report message can have many variables but messages are parsed to the interface database so that one row in the Received_report –table has only one variable.

9.4.4 Creating variables and history tables with *Variable Configuration* –tool

Variable cross-linking table, receivers phonebook and history table may be created by using Variable Configuration tool, which is included with the project (IWSDemo-directory\DB2SMS\DBConfiguration.exe). Please note that the changes happen in real time. If you do a mistake, you cannot cancel!

1. Start the program. You should see a login dialog:

TESTING FTP DEMO



FF-Automation - Variable configuration

FF-Automation - History table configuration

Server: localhost

Username: postgres

Password:

History database name: smshistory

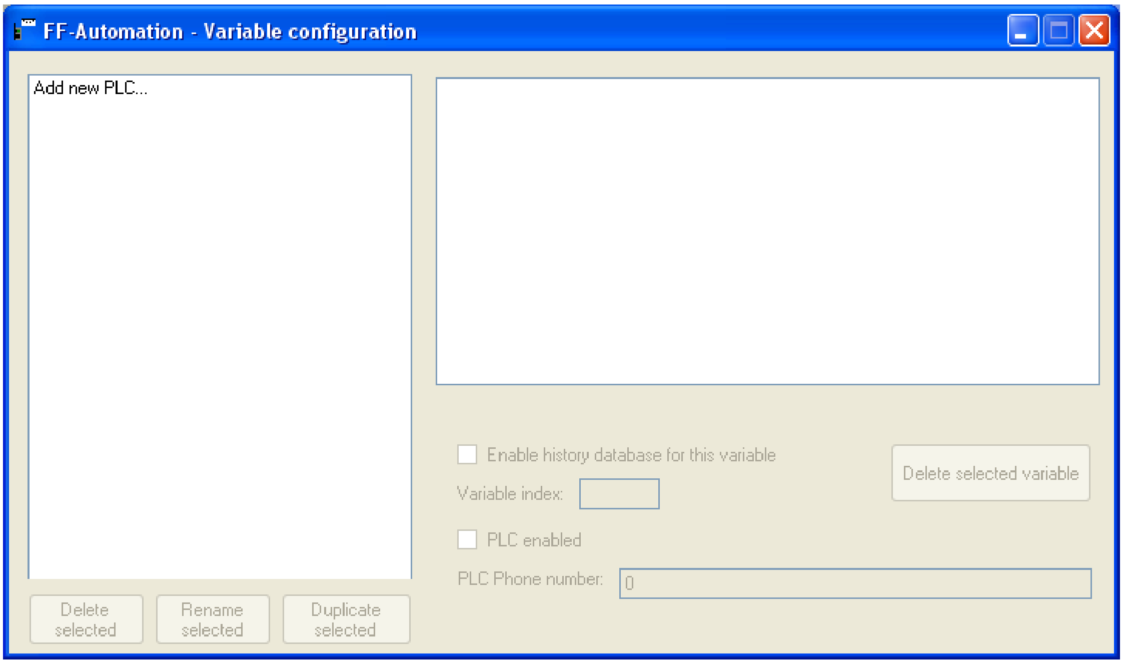
Interface database name: smsindusoft

□ Create history database

Default schema name: smshistory

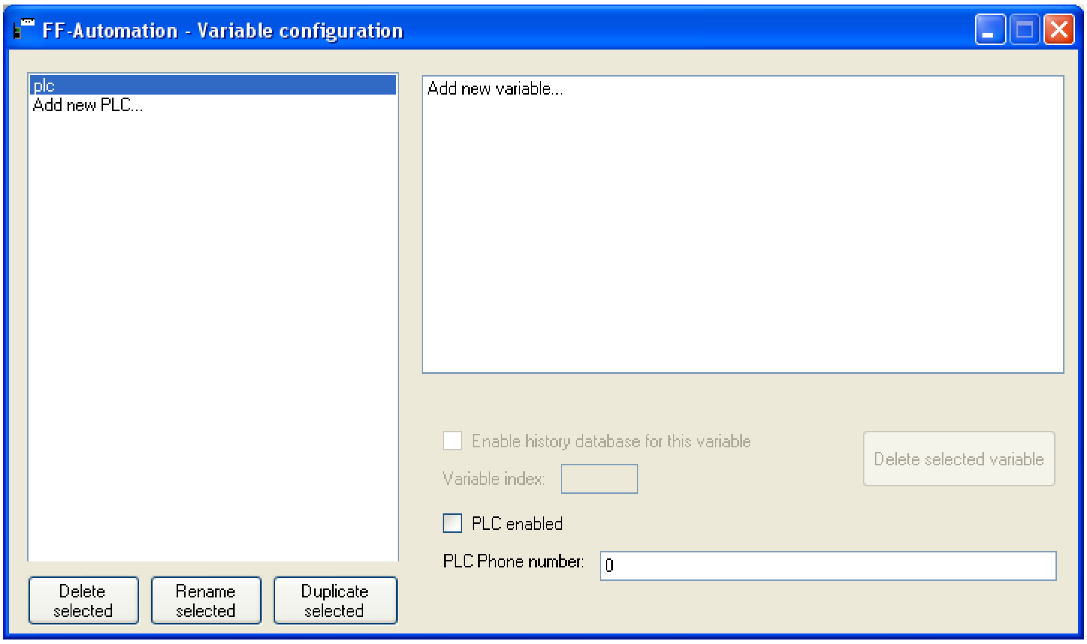
Owner of new tables: iws

2. Write password for user postgres. If you don’t have a history database yet, you may create it by checking checkbox *Create history database*. Normally you don’t need to change *Default schema name* value Click *Continue* when you’re done.
3. After a while, the program opens the main window:



FF-Automation - Variable configuration

4. By double clicking *Add new PLC…* you may create a new PLC.


FF-Automation - Variable configuration

plc
Add new PLC...

Add new variable...

□ Enable history database for this variable

Variable index:

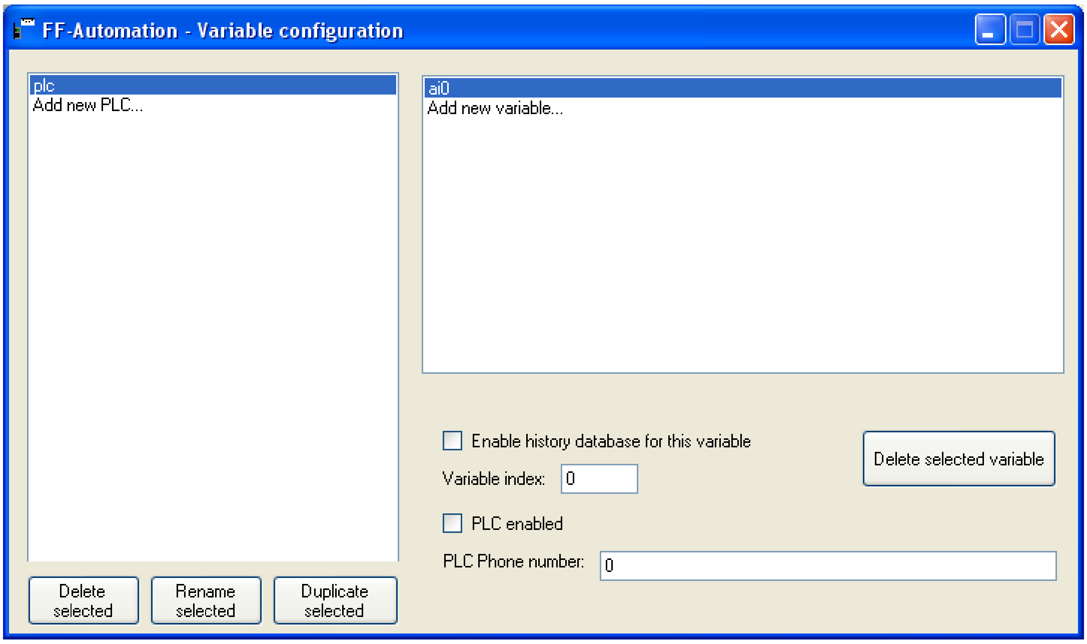
□ PLC enabled

PLC Phone number: 0

5. You may change PLC properties by selecting it from the list. You can remove it (*Delete selected*), rename it (*Rename selected*) or create duplicates of it (*Duplicate selected*). Please note that duplicating creates also new history tables.

   You can also change the phone number of the PLC or enable/disable it. By double clicking *Add new variable…* you may create a new variable for the selected PLC.

6. After you have created a new variable and selected it, you may create a history database for it. Just check *Enable history database for this variable* –checkbox. *Variable index* defines the index of the selected variable while receiving history or report messages.


FF-Automation - Variable configuration

plc
Add new PLC...

ai0
Add new variable...

□ Enable history database for this variable

Variable index: 0

□ PLC enabled

PLC Phone number: 0

7. When you are done with creating variables, just close the program.

9.5 Deleting messages

MyTest –demo project deletes old (flag =0) messages from the interface database –tables ~24 hours period. You can delete the old messages from the table by toggling the following tag values. (Toggling means that 1 turns to 0 or 0 turns to 1. Toggling doesn't mean that 1 turns to 2.)
:

- DBC_ReceiveEvent.Delete
- DBC_ReceiveReport.Delete
- DBC_ReceiveHistory.Delete
- DBC_SendAll.Delete
- DBC_SendSingle.Delete

Program is designed so that it will automatically select the deleted data using one minute period (Look ODBC task).

9.6 Sending error messages

IWS changes the value of the interface database's *Send_Watchdog* –table every 8 seconds. If DB2SMS detects that this value is not changing in defined time, it sends the defined SMS message to the defined phone number. From user interface you can change the error message and the phone number. You may also disable the error message sending.

10 How the IWS links to DB2SMS?

This chapter describes how the communication interface between IWS and the interface database is created. IWS uses ODBC –task to communicate with the interface database. To make the communication automatic, IWS uses scheduler and math tasks. Normally scheduler starts the certain communication process using e.g. time triggering. Math tasks are used to make more complicated scripts. Be sure that you fully understand the functionality of these scripts before you make any changes.

10.1 Sending SMS -message

This project demonstrates only the simple SMS sending process and it works only manually from the user interface. See chapter 5.2 for more information.

10.2 Reading ErrorMessage

ErrorMessage-table is read automatically once per second using Scheduler –task #1.

Clock		00:00:01	DBC_ErrorMessage.Select	NOT DBC_ErrorMessage.Select

This will start the ODBC –task called “Read DB2SMS's state from database”. The ErrorMessage is stored to tag *DB_ErrorMessage.*

10.3 Reading incoming messages from the interface database

Reading incoming SMS messages from the interface database is divided to three phases. This process reads all events (which flag =1). After first event is read it writes its flag to zero. Then it continues to read next event (which flag =1). When all the events are read, all the event flags are zero. Note that events are not deleted from the interface database after reading. Deleting events is a separated process.

Reading Events, reports and history is about similar process.

10.3.1 Phases

- Beginning (*x_Receiver.Begin_MTrig*)
  - In the beginning all the flag=1 arrays are selected from the database. Some tags are initialized.
- 1. phase (*x_Receiver.State0_MTrig*)
  - The event is read and SendEvent –function stores event to the event database. Alarm events are sent to GSM phones.
- 2. phase (*x_Receiver.State1_MTrig*)
  - The events flag is turned to zero.
- 3. phase (*x_Receiver.State2_MTrig*)
  - The next array is selected from the database and the process goes back top phase 1.
- Finish (*x_Receiver.Finish_MTrig*)
  - The Finish process just ends the event receiving process.

10.3.2 ODBC tasks

Task name	Table	Condition	Description
1. Send SMS messages to everyone	Send_All		Send SMS messages to everyone
2. Send single SMS Messages	Send_One		Send single SMS Messages
21. Receive an event message	Received_Events	Flag=1	Receive an event messages
4. The phonebook for sending	Phonebook_Sending	Name={Send_Phones_Name}	Modifying phonebook (group messages)
6. Modify the event table	Received_Events	mindex = {DB_ReceiveEvent.Index}	Modify the event table. Modifies the event’s flag to zero after reading.
7. Receive a history messages	Received_History	Flag=1	Receive a history messages

There are 17 ODBC tasks in the MyTest program:

8. Modify a history table	Received_History	mindex = {DB_ReceiveHistory.Index}	Modify the history table. Modifies the history array's flag back to zero after reading.
9. Save a history column to the history db	Trend_{sender}_{var}		History data storing to thehistory database. Note that the DSN is WS_Hispool, not SMSIndusoft.
10. Receive report messages	Received_Report	Flag=1	Receive report messages
11. Modify report table	Received_Report	mindex = {DB_ReceiveReport.Index}	Modify the report table. Modifies the report array's flag back to zero after reading.
13. Read DB2SMS's state from database	ErrorMessage		Read DB2SMS's state from database
14. Remove old history messages	Received_History	Index={Crosslink_ID}	Remove old history messages
15. Remove old event messages	Received_Event	Index={Crosslink_ID}	Remove old event messages
16. Remove old report messages	Received_Report	Index={Crosslink_ID}	Remove old report messages
17. Remove old send_all messages	Send_All	Index={Crosslink_ID}	Remove old send_all messages
18. Remove old send_single messages	Send_One	Index={Crosslink_ID}	Remove old send_single messages
19. Watchdog alarm resetting	Send_Watchdog	"Name"='IWS' AND "Enabled"=1	Watchdog alarm resetting. (Watchdog should be active and name should be IWS)

10.3.3 Scheduler tasks

For every message type, there is own receiver scheduler –task. These tasks are about identical for all message types, only the tag names differs. This table represents the scheduler task, which controls the event receiving process.

Clock		00:00:05	Event_Receiver.Begin_Mtrig	1	Event_Receiver.Active=1
Change	DBC_ReceiveHistory.Transaction		Event_Receiver.State0_Mtrig	(DBC_ReceiveEvent.Status0=0) AND (Event_Receiver.State=0) AND (Event_Receiver.State2_MTrig = 0) AND Event_Receiver.Begin_MTrig = 0) AND (NOT (DB_ReceiveEvent.Phone = ""))	Event_Receiver.Active=0
Change	DBC_ReceiveHistory.MTransaction		Event_Receiver.State1_Mtrig	(DBC_ReceiveEvent.Status0=0) AND (DBC_ReceiveEvent.Status1=0) AND (Event_Receiver.State=1) AND (Event_Receiver.State0_MTrig = 0)	Event_Receiver.Active=0
Change	DBC_ReceiveHistory.MTransaction		Event_Receiver.State2_Mtrig	(DBC_ReceiveEvent.Status0=0) AND (DBC_ReceiveEvent.Status1=0) AND (Event_Receiver.State=2) AND (Event_Receiver.State1_MTrig = 0)	Event_Receiver.Active=0
Change	DBC_ReceiveHistory.Transaction		Event_Receiver.Finish_Mtrig	(NOT (DBC_ReceiveEvent.Status0=0)) OR (NOT (DBC_ReceiveEvent.Status1=0)) OR (DB_ReceiveEvent.Time="")	Event_Receiver.Active=0

First row activates the initialization. ODBC task is triggered from math function. Second row activates the saving process of the read event to the history database. This phase is activated only is there are no errors in ODBC read (*Event_Receiver.Status0=0*) and program is in the first phase (*Event_Receiver.State=0*) and third phase is not active (*Event_Receiver.State2_MTrig = 0 and initialization is not active* (*Event_Receiver.Begin_MTrig = 0*) and the read event array was not empty (*NOT (DB_ReceiveEvent.Phone = "")*).

3. and 4. rows are similar to 2. row.

The last row activates the Finishing process if something went wrong *NOT (DBC_ReceiveEvent.Status0=0)) OR (NOT (DBC_ReceiveEvent.Status1=0))*) or there are no more new events (*DB_ReceiveEvent.Time=""*).

10.3.4 Math tasks

There are five Math tasks per message type. This represents the Events Math tasks, but Report and history maths are quite similar and only phase 1 differs

10.3.4.1 Initialization (*Event_Receiver.Begin_MTrig*)

Event_Receiver.Active	1	Event receiving flag is turned active
DB_ReceiveEvent.Phone	""	Clearing phone number
DBC_ReceiveEvent.Status0	0	Clearing ODBC error status
DBC_ReceiveEvent.Status1	0	Clearing ODBC error status
Event_Receiver.State0_MTrig	0	Deactivates phase 1
Event_Receiver.State1_MTrig	0	Deactivates phase 2
Event_Receiver.State2_MTrig	0	Deactivates phase 3
Event_Receiver.State	0	Activates the first state
DBC_ReceiveEvent.Select	NOT DBC_ReceiveEvent.Select	Stars ODBC task, Selects all events, which flag = 1
Event_Receiver.Begin_MTrig	0	This math will not be executed again

10.3.4.2 Phase 1 (*Event_Receiver.State0_MTrig*)

	SendEvent(DB_ReceiveEvent.Message + " (" + DB_ReceiveEvent.Date + "/" + DB_ReceiveEvent.Time + ", " + DB_ReceiveEvent.Phone + " (" + DB_ReceiveEvent.Name + "))")	Stores the event to the history database
DB_SendAll.Message	DB_ReceiveEvent.Phone + " (" + DB_ReceiveEvent.Name +"): " + DB_ReceiveEvent.Message	Copies the event message, for group SMS purposes
DB_SendAll.Flag	1	Group message flag is set to 1
DBC_SendAll.Insert	If(StrLeft(StrLower(DB_ReceiveEvent.Message),5)="alarm",NOT DBC_SendAll.Insert,DBC_SendAll.Insert)	If the event message starts with the word alarm or ALARM, then it will be send using group SMS sending.
DBC_ReceiveEvent.Mselect	NOT DBC_ReceiveEvent.Mselect	Selects array for modification.
Event_Receiver.State	1	Moves to phase 2
Event_Receiver.State0_MTrig	0	This math will not be executed again

10.3.4.3 Phase 2 (*Event_Receiver.State1_MTrig*)

DB_ReceiveEvent.Flag	0	Flag is turned to zero (array is read)
DBC_ReceiveEvent.MUpdate	NOT DBC_ReceiveEvent.MUpdate	The flag field is updated to zero.
Event_Receiver.State	2	Next phase
Event_Receiver.State1_MTrig	0	This math will not be executed again

10.3.4.4 Phase 3 (Event_Receiver.State2.Mtrig)

DB_ReceiveEvent.Phone	""	Phone number is cleared
DBC_ReceiveEvent.Next	NOT DBC_ReceiveEvent.Next	Next event is read
Event_Receiver.State	0	Back to phase 1
Event_Receiver.State2_MTrig	0	This math will not be executed again

10.3.4.5 Finish (*Event_Receiver.Finish_MTrig*)

Event_Receiver.Active	0	Receiving has Finished
Event_Receiver.Finish_MTrig	0	This math will not be executed again

10.3.5 Report and history maths (1. phase only)

10.3.5.1 History receiving math 1. Phase (*History_Receiver.State0_MTrig*)

DB_HistoryPool.Variable	DB_ReceiveHistory.Name + "_" + DB_ReceiveHistory.Variable	This sets the correct table name in the MySQL database
DB_HistoryPool.TimeStamp	DB_ReceiveHistory.Date + " " + DB_ReceiveHistory.Time	This sets the timestamp
DB_HistoryPool.Value	DB_ReceiveHistory.Value	This sets the value
DBC_HistoryPool.Insert	NOT DBC_HistoryPool.Insert	The data is inserted to MySQL database
DBC_ReceiveHistory.Mselect	NOT DBC_ReceiveHistory.Mselect	The history array is selected for modification. (Flag will be set to zero)
History_Receiver.State	1	Next phase
History_Receiver.State0_MTrig	0	This math will not be executed again

10.3.5.2 Report_receiving 1. Phase (Report_*Receiver.State0_MTrig*)

Report_Pointer	DB_ReceiveReport.Name + "." + DB_ReceiveReport.Variable	Pointer gets correct class variable's name: (Sender's) Name.Variable
@Report_Pointer	DB_ReceiveReport.Value	Value is set to pointed variable
DBC_ReceiveReport.MSelect	NOT DBC_ReceiveReport.MSelect	The report array is selected for modification. (Flag will be set to zero)
Report_Receiver.State	1	Next phase
Report_Receiver.State0_MTrig	0	This math will not be executed again

10.4 Removing old (flag = 0) messages

IWS scheduler task *Remove old messages from the interface database* is configured to remove automatically all the old messages (flag =0) in 23:59:59 (~1 day) intervals. Note that new messages (not read messages) has flag =1, so these won't be removed.

Calendar	23:59:59	DBC_ReceiveEvent.Delete	NOT DBC_ReceiveEvent.Delete
Calendar	23:59:59	DBC_ReceiveReport.Delete	NOT DBC_ReceiveReport.Delete
Calendar	23:59:59	DBC_ReceiveHistory.Delete	NOT DBC_ReceiveHistory.Delete
Calendar	23:59:59	DBC_SendAll.Delete	NOT DBC_SendAll.Delete
Calendar	23:59:59	DBC_SendSingle.Delete	NOT DBC_SendSingle.Delete

10.5Watchdog timer

DB2SMS sends SMS message to the defined phone number if it is doesn’t get “heartbeat” signal from IWS. This can happen if IWS program is jammed. IWS updates the interface database’s watchdog using *Reset watchdog Counter*-scheduler task.

Clock		00:00:08		DB_Watchdog.Flag	DB_Watchdog.Enabled	
Clock		00:00:08		DB_Watchdog.PRG_Counter	Second	
Clock		00:00:08		DB_Watchdog.Old_PRG_Counter	DB_Watchdog.PRG_Counter - 1	
Clock		00:00:08		DB_Watchdog.A2SMS_Counter	0	
Clock		00:00:08		DBC_Watchdog.Update	NOT DBC_Watchdog.Update	

11 Tags and class tags

Tag	Type	Description
DB_SendAll	Send_All	Send_All –tables active arrays values
DB_SendSingle	Send_Single	Send_One – tables active arrays values
DB_ReceiveEvent	Rec_Event	Received_Events – table's active arrays values
DB_ReceiveHistory	Rec_History	Received_History – table's active arrays values
DB_ReceiveReport	Rec_Report	Received_Report – table's active arrays values
DB_HistoryPool	HisPool	History database's active arrays values
DB_ErrorMessage	String	ErrorMessage – table's active arrays values
DBC_SendAll	ODBC_MsgSender	Send_All –table control (ODBC task)
DBC_SendSingle	ODBC_MsgSender	Send_One – table control (ODBC task)
DBC_ReceiveEvent	ODBC_Receivers	Received_Events – table control (ODBC task)
DBC_ReceiveHistory	ODBC_Receivers	Received_History – table control (ODBC task)
DBC_HistoryPool	ODBC_HisPool	History – database control (ODBC task)
DBC_ReceiveReport	ODBC_Receivers	Received_Report – table control (ODBC task)
DBC_ErrorMessage	ODBC_ErrorMessage	ErrorMessage – table control (ODBC task)
DBC_Send_Phones	ODBC_Phonebook	Phonebook_Sending – table control (ODBC task)
DB_Send_Phones	Phonebook	Phonebook_Sending – table's active arrays values
DBC_Receive_Phones	ODBC_Phonebook	Phonebook_Receiving – table control (ODBC task)
DB_Receive_Phones	Phonebook	Phonebook_Receiving – table's active arrays values
Event_Receiver	Receiver	Event receiver class variables
History_Receiver	Receiver	history receiver class variables
Report_Receiver	Receiver	Report receiver class variables
DBC_Crosslink	ODBC_Phonebook	Report_Crosslinks - table control (ODBC task)
DB_Crosslink	Crosslink	Report_Crosslinks – table's active arrays values
Iface_String0	String	Interface variable
Iface_String1	String	Interface variable
Iface_String2	String	Interface variable
Iface_String3	String	Interface variable
Iface_String4	String	Interface variable
Iface_String5	String	Interface variable
Iface_String6	String	Interface variable
Iface_String8	String	Interface variable
Iface_String7	String	Interface variable
Iface_History_DataGrid_Update	Boolean	History data update after selecting variable name
Iface_History_Grid_Update	Boolean	History variable grid update after selecting plc
Iface_PB_Sending_Grid_Update	Boolean	Outgoing phonebook update after editing it in interface
Iface_Phonebook_receiving_Update	Boolean	PLC data read after selecting PLC from a grid
Send_Phones_Name	String	Selected outgoing phone number's name
AI1Alarm	Boolean	Alarm indication tag for AI1
Report_Pointer	String	Report pointer tag which gets the class variable's name
PLC[10]	PLC	Report variables for PLCs
DBC_Watchdog	ODBC_Watchdog	Watchdog timer's table control (ODBC task)
DB_Watchdog	Watchdog	Watchdog timer – table's active arrays values
WD_ResetTrig	Boolean	Resetting tag for watchdog timer
DB_Config_DSN	String	Interface database DSN
DB_Config_Password	String	Interface database password
DB_Config_Status	String	Interface database status
DB_Config_Username	String	Interface database username
HisPool_DSN	String	History database DSN
HisPool_Password	String	History database password
HisPool_Username	String	History database username

Class	**Tag (member)**	**Type**	**Description**
Crosslink			Crosslink class variable
	ID	Integer	Index
	Name	String	Name
HisPool			History data class variable (history data saving)
	Value	String	
	TimeStamp	String	
	TimeStamp_ms	Integer	milliseconds
	Variable	String	
ODBC_ErrorMessage			Error message class variable
	Select	Boolean	Select trigger
ODBC_HisPool			History data ODBC control class variable
	Select	Boolean	Select trigger
	Insert	Boolean	Insert trigger
	Next	Boolean	Next trigger
ODBC_MsgSender			Message sender class variable (both Group and single)
	Insert	Boolean	Insert trigger
	Select	Boolean	Select trigger
	Delete	Boolean	Delete trigger
ODBC_Phonebook			Phonebook data ODBC control class variable
	Delete	Boolean	Delete trigger
	Insert	Boolean	Insert trigger
	Select	Boolean	Select trigger
	Complete	Boolean	Transaction Complete
ODBC_Receivers			Reveiving message ODBC control class variable
	Status0	Integer	Status
	Select	Boolean	Select trigger
	Next	Boolean	Next trigger
	MSelect	Boolean	Select trigger (DB modification)
	MUpdate	Boolean	Update trigger (DB modification)
	Status1	Integer	Status (DB modification)
	Delete	Boolean	Delete trigger
Phonebook			Phonebook class variable
	Phone	String	Phone number
	Enabled	Boolean	Active flag
	Name	String	Name
PLC			PLC class variable (example)
	AI0	Integer	Analog input 0
	AI1	Integer	Analog input 1
	AI2	Integer	Analog input 2
	AI3	Integer	Analog input 3
	WM0	Integer	Word Memory 0
	WM1	Integer	Word Memory 1
	WM2	Integer	Word Memory 2
	WM3	Integer	Word Memory 3
	Temperature	Real	Scaled temperature
Rec_Event			Receiving event messages class variable
	Flag	Integer	”New message” –flag
	Date	String	date
	Time	String	time
	Phone	String	Phone number
	Message	String	message
	Name	String	sender’s name (read from phonebook by DB2SMS)
	Index	Integer	Database index of the received message
Rec_History			Receiving history messages class variable
	Flag	Integer	”New message” –flag
	Date	String	date
	Time	String	time
	Phone	String	Phone number
	Variable	String	Variable name
	Value	String	value

	Name	String	sender’s name (read from phonebook by DB2SMS)
	Index	Integer	Database index of the received message
Rec_Report		Integer	Receiving report messages class variable
	Flag	String	”New message” –flag
	Date	String	date
	Time	String	time
	Phone	String	Phone number
	Variable	String	Variable name
	Value	String	value
	Name	String	sender’s name (read from phonebook by DB2SMS)
	Index	Integer	Database index of the received message
Receiver			Message receiver math control class variable
	Active	Boolean	Receiver is active
	Begin_MTrig	Boolean	Beginning of receiving math trigger
	State0_MTrig	Boolean	1. phase math trigger
	State1_MTrig	Boolean	2. phase math trigger
	State2_MTrig	Boolean	3. phase math trigger
	Finish_MTrig	Boolean	Finishing math trigger
	State	Integer	Receiving state
Send_All			Send_All – table ODBC control class variable
	Flag	Integer	”send message” -flag
	Message	String	Message
Send_Single			Send_One –table
	Flag	Integer	”send message” -flag
	Phone	String	Phone number
	Message	String	Message
ODBC_Watchdog			Watchdog timer ODBC control class variable
	Select	Boolean	Select trigger
	Update	Boolean	Update trigger
Watchdog			Watchdog timer class variable
	Name	String	Watchdog name
	Enabled	Integer	activation
	Flag	Integer	message send (0=send, 1=not send yet)
	PRG_Counter	Integer	This should be changed
	Old_PRG_Counter	Integer	Old value
	A2SMS_Counter	Integer	Error status counter
	Counter_Max	Integer	Error status timeout
	Phone	String	Phone number
	Message	String	Message

Part 4 – AutoLog GSM-RTU Demo application

You should now read the following manual:

“AutoLog_GSM_PLC_Service_Manual”

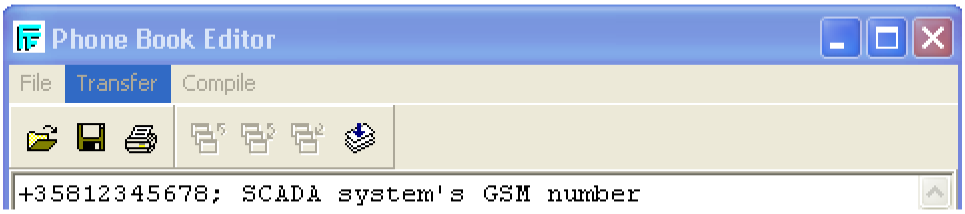
to learn how to make application programs for AutoLog GSM RTUs. After that you should be able to understand how the GSM pump station monitoring demo program works and how you can modify it to make your own special application.

11.1 Example Source code

```
; GSM_PUMP_MONITORING DEMO PROGRAM
; VERSION 1.01 DATE 2nd March 2015
; AUTHOR: AM / FF-Automation Oy
; Water pump station monitoring only, program doesn't control pumps, ask other demo if
you need pump controlling!
; For DEVICE: AutoLog GSM-8, AutoLog GSM-16 or AutoLog GSM-20
; Pump start count and operating times, surface level and digital input status and
alarms
; Indusoft Web Studio SCADA with DB2SMS GSM driver using PostgreSql database interface

; List of all used inputs, outputs and memories in this program
; ------------------------------------------------------------
; AI0 = water level (4-20mA->0-4 m)
; AI1 = water pressure (4-20mA->0-10 bar)
; AI2 = water pumps station temperature (Pt100->-50..+150°C), note that GSM-8 has only
2 analog inputs

; DI0 = High water level limit switch (NC= Normally Closed), 0 = alarm , 1 = normal
state, DI0=!M0
; DI1 = Pump 1 heat relay alarm (NC),
; DI2 = Pump 1 control switch state, Manual (0) / Auto (1)
; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped , DI3=M3

; M0 = ...
; M1 = ...

; WM50 = for initialization condition

; Contents of this program
; -------------------------------------------------------------
; 0.0 Version
; 0.1 Initializations
; ....


; PROGRAM STARTS HERE
;**************************************************************************

; 0.0 Version
; ------------------------------
'M50#!M50' AO222=100 ; Version number can be placed e.g. on AO222 variable, which is
hardly used in other purposes.
; It is recommended that you define variables for program type numbers if you make many
variants

; Empty lines reserved for program updates (without changing the line numbers of other
programming lines)
; These lines are useful if you need to download program changes remotely through GSM
network later on.

.empty
.empty
.empty
.empty
.empty

; 0.1 INITIALISATIONS
;-------------------------------------------------------------------------------
; M2 = Pump 1 Manual (0) / Auto (1)= DI2
'DI2' M2  ;Pump 1 Manual (0) / Auto (1) state to (binary memory number 2) M2
```

```
'M50#!M50' WM50=AI0 WM51=AI1 WM52=AI2 WM53=AI3 ; analog inputs raw values are copied to
WM50-53

; 0.2 (A) GSM modem is resetted at clock 2359 FOR AUTOLOG GSM-4, GSM-8, GSM-16
;-------------------------------------
'CLK=2359' M90
'M90=1' RO30=1 ;resets gsm modem, system program sets it back to 0 automatically, just
in case the modem gets jammed.
; RO30 resets embedded GSM modem which is used in GSM-4, -8 and -16

; 0.2 (B) GSM modem is resetted at clock 2359 FOR AUTOLOG GSM-20 with external GSM
modem
;-------------------------------------
; GSM-20 has external GSM modem which needs to be resetted using external relay which
is controlled by digital output
; 'M90=1' DO0=1 ;resets GSM-20 GSM modem. Conennect DO0 to external relay to cut off
the power from the GSM modem.
; 'DO0S10' DO0=0


; 1.0 DIGITAL ALARMS
;---------------------------------------------------------------------------
; 1.1 HIGH LEVEL ALARM AND NORMALIZATION
;---------------------------------------------------------------------------
; DI0 = High water level limit switch (NC= Normally Closed), 0 = alarm , 1 = normal
state, DI0=!M0
; M0 = High water level alarm bit, value: 1 = alarm (DI0=0) , 0 = normal state (DI0=1)

'!DI0S3' M0=1                 ; ALARM state, 3 seconds delay for water waving, note: NC
input!
'DI0S120' M0=0                ; NORMALIZATION state, 120 seconds delay, for hysteresis
(so that the alarm state doesn't repeat too often)
.empty                        ; Empty lines are reserved for program updates through
sms.
.empty

; 1.2 HEAT RELAY ALARM (PUMP 1 FAULT) AND NORMALIZATION
;---------------------------------------------------------------------------
; DI1 = Pump 1 heat relay alarm (NC),
; M1 = Pump 1 heat relay alarm bit, value: 1 = alarm (DI1=0) , 0 = normal state (DI1=1)

'!DI1S3' M1=1                                     ; ALARM state 1 second delay "just in
case", note: NC tulo!
'DI1S120' M1=0                                    ; NORMALIZATION delay 120 seconds for
"hysteresis"
.empty

; 2.0 ANALOG ALARMS
;---------------------------------------------------------------------------
; 2.1 HIGH PRESSURE ALARM
;---------------------------------------------------------------------------
; AI1 = water pressure (4-20mA->0-10 bar)
; WM51 = AI1
; M10 = high pressure alarm bit
; M47 = condition variable

'WM51<800' M47 ; if under 2 bars over 30 minutes (scaling for analog inputs 0-10 bar
raw value is 0-4000 so 2 bar=800)
'M47M30' M10=1 ; if Memort47 is TRUE over 30 minutes then M10=1
'!M47M30' M10=0 ; if Memort47 is NOT TRUE over 30 minutes then M10=0

.empty
.empty
.empty


;3.0 PUMP 1 (P1) START COUNT AND OPERATING TIME
;---------------------------------------------------------------------------

;3.1 P1 24H START COUNTER (during 24 hour time) and P1 TOTAL START COUNTER
;---------------------------------------------------------------
; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; M3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; WM10 = P1 START COUNTER (today)
; P1 TOTAL START COUNTER
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x
; P1 TOTAL START COUNTER = (10000 x WM13) + WM12 (Counted in SCADA!), Note: WM (Word
Memories) has 16 bits, so the maximum is 65535

'DI3S1' M3
'M3=1' WM10+1 WM12+1
'WM12>9999' WM13+1 WM12=0
.empty

;3.2 P1 24H OPERATING TIME and P1 TOTAL OPERATING TIME
;--------------------------------------------------
; DI3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; WM14 = P1 OPERATING TIME SECONDS (today)
; WM15 = P1 OPERATING TIME MINUTES (today)
; WM16 = P1 OPERATING TIME HOURS (today)
; WM20 = P1 OPERATING TIME SECONDS TOTAL
; WM23 = P1 OPERATING TIME IN SECONDS sssss today

'DI3&P1' WM14+1 WM20+1 WM23+1 ;Seconds (today), TOTAL and Seconds sssss today.
'WM14>59' WM15+1 WM14=0       ;minutes (today)
'WM15>59' WM16+1 WM15=0                 ;hours (today)
.empty

; P1 TOTAL OPERATING TIME
; WM20 = P1 OPERATING TIME SECONDS TOTAL
; WM21 = P1 OPERATING TIME MINUTES TOTAL
; WM22 = P1 OPERATING TIME HOURS TOTAL HHHHH 0-65535 hours

'WM20>59' WM21+1 WM20=0       ;minutes TOTAL
'WM21>59' WM22+1 WM21=0                 ;hours TOTAL
.empty

; 4.0 P1 RESETTING 24H COUNTERS
;------------------------------------------------------
; M80 = CLOCK TRIGGERED VARIABLE
; WM11 = P1 START COUNT LAST 24H HISTORY CLOCK 7:00 - klo 7:00
; WM60 maximum SMS count per day, safety limit

'CLK=0700' M80 ; this is true when PLC's RTC time is 7:00 for one program cycle only
'M80=1' WM11=WM10 WM10=0 WM60=0; P1 start count move and reset, commands are executed
from left to right order.
.empty

; P1 Operating time 24h reset

; WM14 = P1 OPERATING TIME SECONDS (today)
; WM15 = P1 OPERATING TIME MINUTES (today)
; WM16 = P1 OPERATING TIME HOURS (today)

; WM17 = P1 OPERATING TIME SECONDS (last 24h history)
; WM18 = P1 OPERATING TIME MINUTES (last 24h history)
; WM19 = P1 OPERATING TIME HOURS (last 24h history)
; WM23 = P1 OPERATING TIME IN SECONDS sssss today
; WM24 = P1 OPERATING TIME IN SECONDS sssss (last 24h history)

'M80=1' WM17=WM14 WM18=WM15 WM19=WM16 WM24=WM23 WM14=0 WM15=0 WM16=0 WM23=0; P1
operating time move and reset
.empty

; 4.1 START COUNT & OPERATING TIME ALARMS
;---------------------------------------------------------------------------
.empty
.empty   ; These are made in SCADA system
.empty

; 5.0 SMS SENDINGS
;---------------------------------------------------------------------------

; 5.11 PACKING BINARY STATES TO ONE 16-BIT WORD MEMORY AND SENDING

'M50#!M50' WM99=M0  ;SPECIAL COMMAND WHICH PACKS BINARY MEMORIES M0-M15 TO ONE 16-BIT
WORD MEMORY
.empty

; M0 = High water level alarm bit, value: 1 = alarm (DI0=0) , 0 = normal state (DI0=1)
; M1 = Pump 1 heat relay alarm bit, value: 1 = alarm (DI1=0) , 0 = normal state (DI1=1)
; M2 = Pump 1 Manual (0) / Auto (1)= DI2
; M3 = Pump 1 Operating state, 1 = Running 0 = Stopped
; M10 = High presure alarm bit
; WM50-WM53 analog input raw values (0-4000) Note! 12-bit Analog inputs are calibrated
to 0-4000 not 0-4096 for easier handling.

; WM60 maximum SMS count per day, safety limit
; WM11 = P1 START COUNT LAST 24H HISTORY CLOCK 7:00 - klo 7:00
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x

; WM24 = P1 OPERATING TIME IN SECONDS sssss (last 24h history)
; WM22 = P1 OPERATING TIME HOURS TOTAL HHHHH 0-65535 hours

; 5.1 ALARM MESSAGE
; ----------------------------
; Alarm message is send when any of the alarm bit changes its value to alarm state or
normal state.
; After the GSM-PLC is booted, only active alarm generates sms (M0-M127 are reseted
when GSM-PLC boots up)
; Alarm activation delay should be the same for every alarm so after the boot up
maximum 1 alarm sms is send.
; DB2SMS format:                        Dn,,Var(n),Var(n+1),Var(n+2),Var(n+3)
; DB2SMS is driver between SCADA and GSM-PLCs. Look more from DB2SMS documentation!

; ALARM is send when any of the alarm bit changes its value to alarm state or to normal
state.
; DB2SMS format: Dn,[opt timestamp
DDMMYYHHMMSS],Var(n),Var(n+1),Var(n+2),Var(n+3),Var(n+4),Var(n+5)

'(M0=0#M0=1#M1=0#M1=1#M10=0#M10=1)&WM60<10' WM60+1 M110
'M110=1&WM60<10' "D9,,1,%WM99,%WM50,%WM51,%WM52,%WM53,%WM11,%WM12,%WM13,%WM24,%WM22" 0
; D9,,1 one in here informs for SCADA system that this is alarm message so it is not
shown in daily report view
.empty

; 5.2 P1 REPORT MESSAGE (Send once every day at 07:00)
;-------------------------------------------------

; Same information is send as in Alarm message
; DB2SMS format: Dn,[opt timestamp
DDMMYYHHMMSS],Var(n),Var(n+1),Var(n+2),Var(n+3),Var(n+4),Var(n+5)

'M80=1&WM60<10'WM60+1
"D9,,0,%WM99,%WM50,%WM51,%WM52,%WM53,%WM11,%WM12,%WM13,%WM24,%WM22"0 ; P1 tiedot
.empty
; ...."D9,,0 zero in here means that this is report message and it is shown in SCADA in
report view

; 6.0 SET PUMP 1 TOTAL START COUNT & TOTAL OPERATING TIME WITH SMS
;-----------------------------------------------------------------------
; WM12 = P1 TOTAL START COUNTER low byte 0-9999
; WM13 = P1 TOTAL START COUNTER high byte 10000x

'($P1)' M70
'M70=1&WM0<10000' WM12=WM0 WM13=0 WM22=WM1  "SET: P1 TOTAL START COUNT = %WM12, P1
TOTAL OPERATING TIME = %WM22 HOURS" 254
'M70=1&WM0>9999' "SET FAILED!: P1 = START COUNT MAXIMUM is 9999" 254
.empty

; 7.0 REPORT SMS QUERY
;-----------------------------------------------------------------------
'($REPORT)' M72
'M72=1' "StartCount-24h=%WM11,-
Tot=%WM22h,Level=%AI0m,HLimAlarm=%M0,HeatRelay=%M1,Man/Aut=%M2,GsmSign(0-30)=%RO98" 254
.empty

; 8.0 CONTROLLING OUTPUT WITH SMS message
;-------------------------------------------------------------------
'($CONTROL)' M69
'M69=1' DO1S5 "Digital Output 1 is controlled on for 5 seconds." 254
```

11.2 Other needed configurations



```
+35812345678; SCADA system's GSM number
```

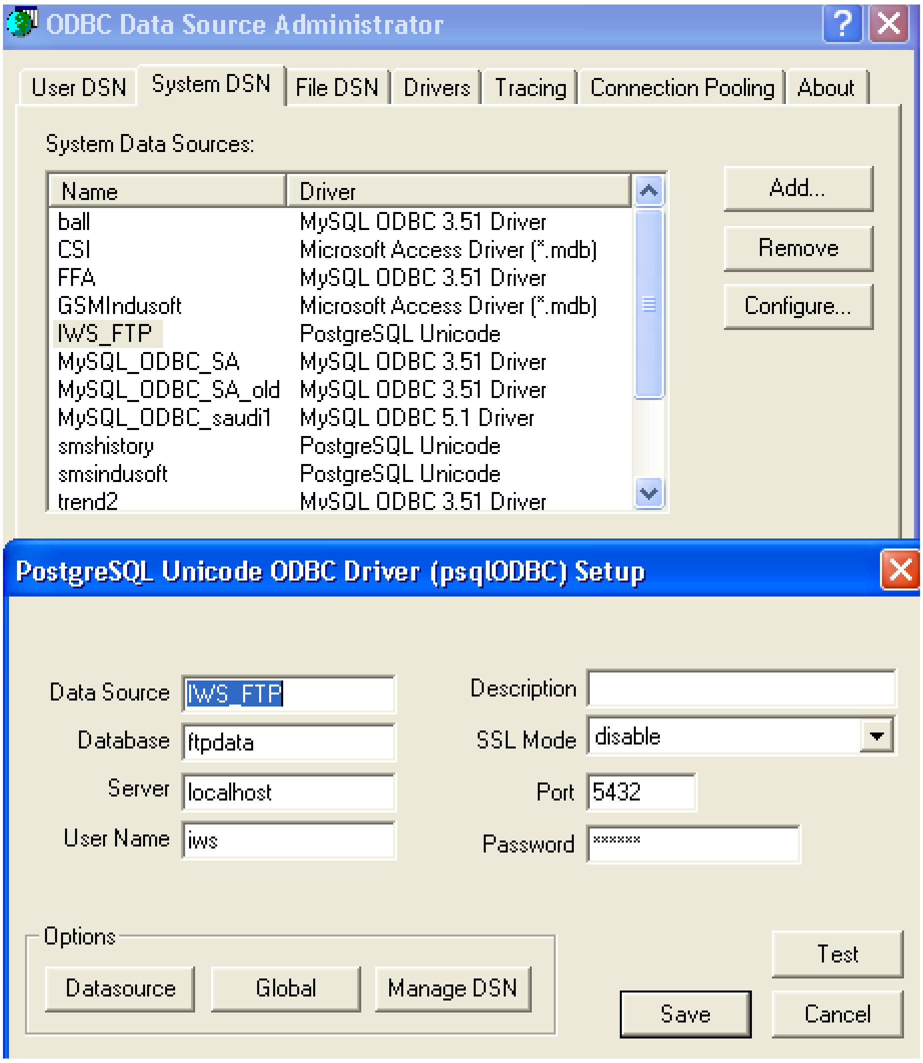
You should define the SCADA systems GSM phone number.

Check that DIP switches and Jumpers are in correct position

When you test the system with GSM modem, disable SIM card's PIN code query, otherwise it will get automatically jammed and asks for PUK code. PIN code query can be disabled by moving the SIM card to normal GSM phone and then go to Settings->Security etc. to disable the PIN code request.

Check other

11.3 Adding ODBC DSN



System Data Sources:

Name	Driver
ball	MySQL ODBC 3.51 Driver
CSI	Microsoft Access Driver (*.mdb)
FFA	MySQL ODBC 3.51 Driver
GSMIndusoft	Microsoft Access Driver (*.mdb)
IWS_FTP	PostgreSQL Unicode
MySQL_ODBC_SA	MySQL ODBC 3.51 Driver
MySQL_ODBC_SA_old	MySQL ODBC 3.51 Driver
MySQL_ODBC_saudi1	MySQL ODBC 5.1 Driver
smshistory	PostgreSQL Unicode
smsindusoft	PostgreSQL Unicode
trend2	MySQL ODBC 3.51 Driver

Data Source: IWS_FTP
Description:
Database: ftpdata
SSL Mode: disable
Server: localhost
Port: 5432
User Name: iws
Password: ******

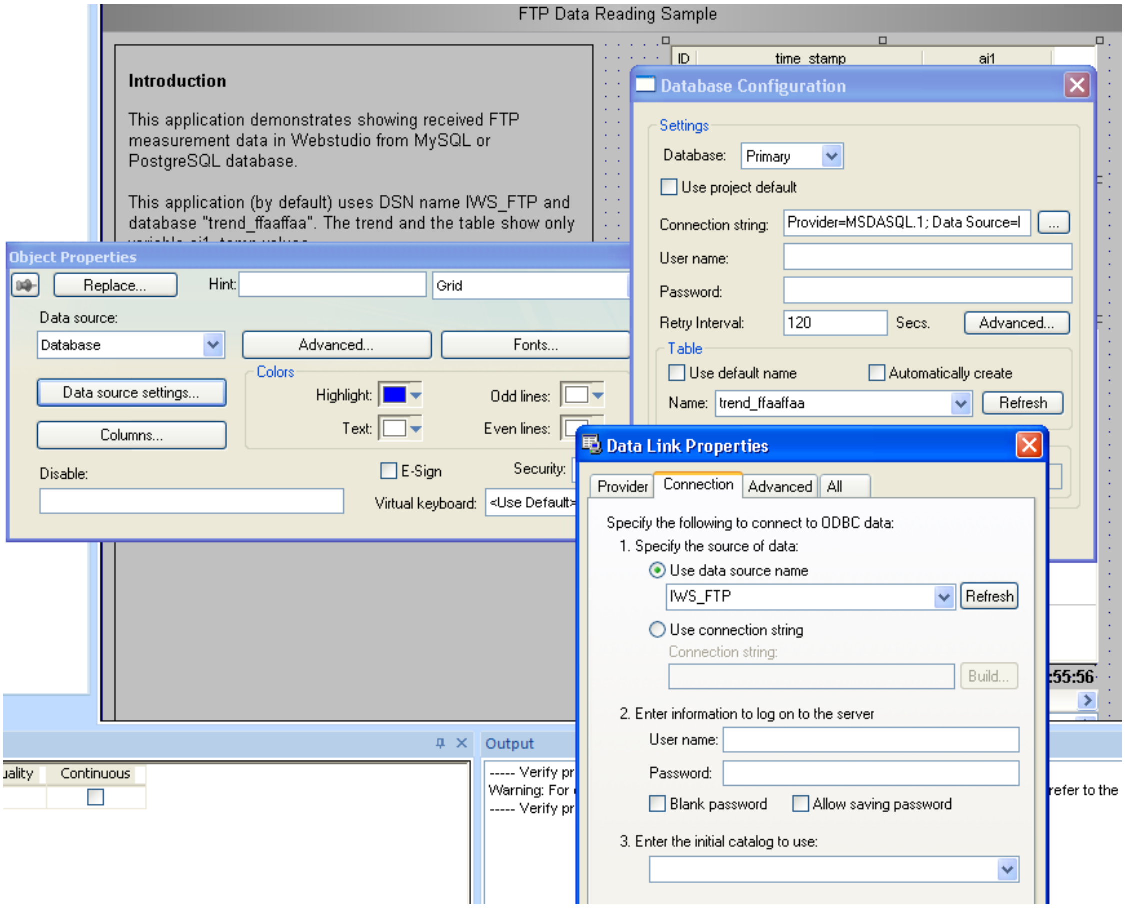
11.3.1 Configuring IWS FTP_parser_DEMO application

You have just created DSN –name ”IWS_FTP” which is used as a link between Indusoft web studio and the “ftpdata” -database

Configuring IWS-project is done simple by changing every database references in the project to use just created DSN-name. These places are:

11.3.1.1 IWS_FTPParser_Demo application:

Appendix



FTP Data Reading Sample

Introduction

This application demonstrates showing received FTP measurement data in Webstudio from MySQL or PostgreSQL database.

This application (by default) uses DSN name IWS_FTP and database "trend_ffaaffaa". The trend and the table show only

Object Properties

Replace... Hint: Grid

Data source: Database

Advanced... Fonts...

Data source settings...

Colors: Highlight: Text: Odd lines: Even lines:

Columns...

Disable:

□ E-Sign Security:

Virtual keyboard: <Use Default>

Database Configuration

Settings

Database: Primary

□ Use project default

Connection string: Provider=MSDASQL.1; Data Source=I ...

User name:

Password:

Retry Interval: 120 Secs. Advanced...

Table

□ Use default name □ Automatically create

Name: trend_ffaaffaa Refresh

Data Link Properties

Provider | Connection | Advanced | All

Specify the following to connect to ODBC data:

1. Specify the source of data:

● Use data source name

IWS_FTP Refresh

○ Use connection string

Connection string: Build...

2. Enter information to log on to the server

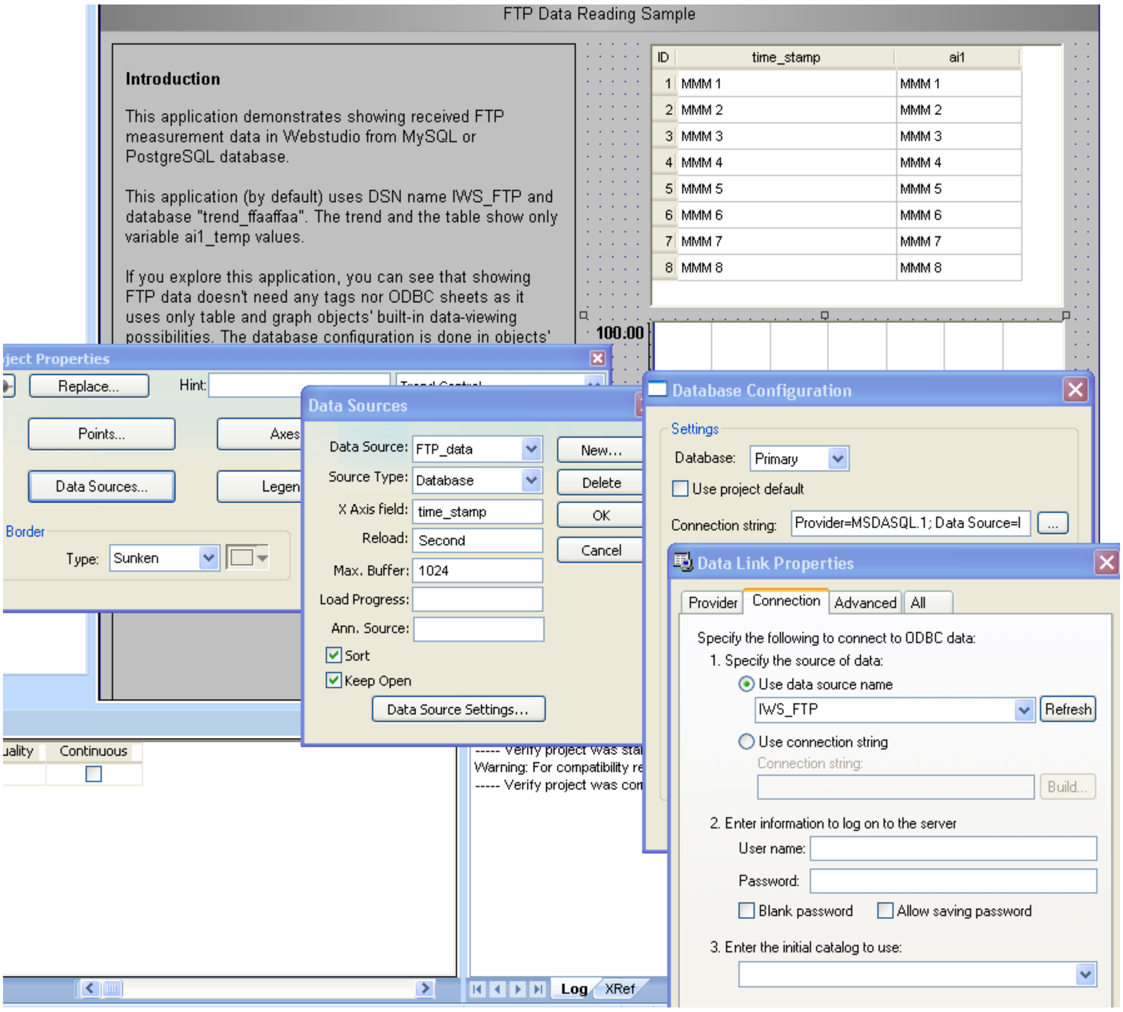
User name:

Password:

□ Blank password □ Allow saving password

3. Enter the initial catalog to use:

Double click “grid object” ->Select Data source settings->Select “…” on the right side of the connection string->Select Connection –tab->



FTP Data Reading Sample

Introduction

This application demonstrates showing received FTP measurement data in Webstudio from MySQL or PostgreSQL database.

This application (by default) uses DSN name IWS_FTP and database "trend_ffaaffaa". The trend and the table show only variable ai1_temp values.

If you explore this application, you can see that showing FTP data doesn't need any tags nor ODBC sheets as it uses only table and graph objects' built-in data-viewing possibilities. The database configuration is done in objects'

ID	time_stamp	ai1
1	MMM 1	MMM 1
2	MMM 2	MMM 2
3	MMM 3	MMM 3
4	MMM 4	MMM 4
5	MMM 5	MMM 5
6	MMM 6	MMM 6
7	MMM 7	MMM 7
8	MMM 8	MMM 8

Type: Sunken

Data Sources

Data Source: FTP_data
Source Type: Database
X Axis field: time_stamp
Reload: Second
Max. Buffer: 1024
Load Progress:
Ann. Source:
☑ Sort
☑ Keep Open

Database Configuration

Settings
Database: Primary
☐ Use project default
Connection string: Provider=MSDASQL.1; Data Source=I

----- Verify project was sta
Warning: For compatibility re
----- Verify project was con

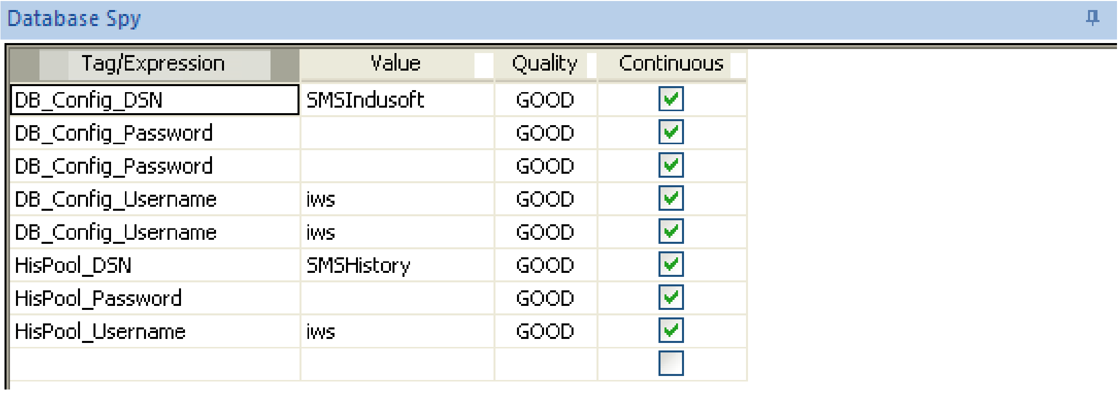
Data Link Properties

Specify the following to connect to ODBC data:
1. Specify the source of data:
● Use data source name
IWS_FTP
○ Use connection string
Connection string:
2. Enter information to log on to the server
User name:
Password:
☐ Blank password ☐ Allow saving password
3. Enter the initial catalog to use:

Double click “trend object” ->Select Data source settings->Select “…” on the right side of the connection string->Select Connection –tab->

11.3.2 DB2SMS driver IWS application:

- *DB_Config_DSN* Startup-value
    - Please notice, that following variables should values be nothing
        - DB_Config_Username
        - DB_Config_Password



Database Spy

Tag/Expression	Value	Quality	Continuous
DB_Config_DSN	SMSIndusoft	GOOD	☑
DB_Config_Password		GOOD	☑
DB_Config_Password		GOOD	☑
DB_Config_Username	iws	GOOD	☑
DB_Config_Username	iws	GOOD	☑
HisPool_DSN	SMSHistory	GOOD	☑
HisPool_Password		GOOD	☑
HisPool_Username	iws	GOOD	☑
			☐

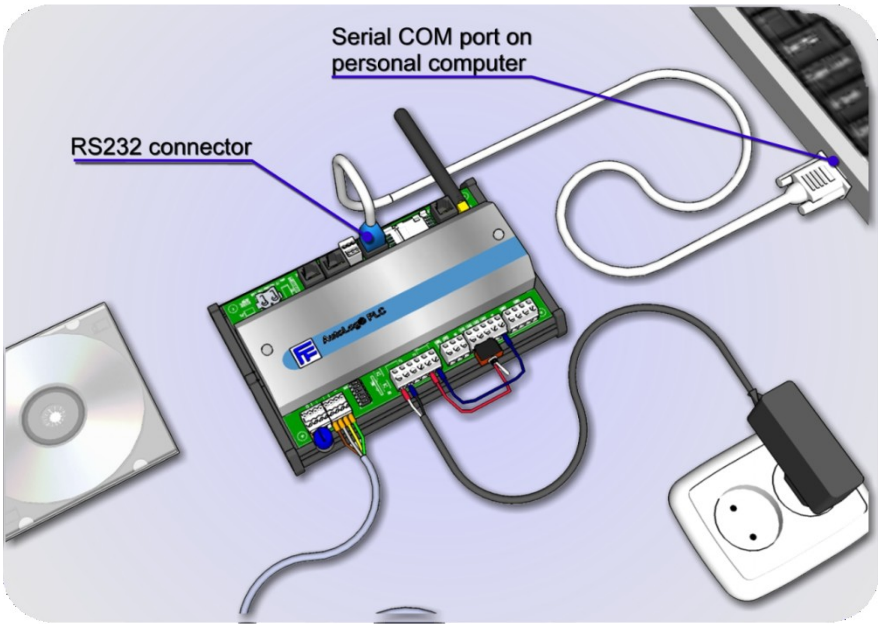
- Lists of *Crosslinktable* -page

Appendix

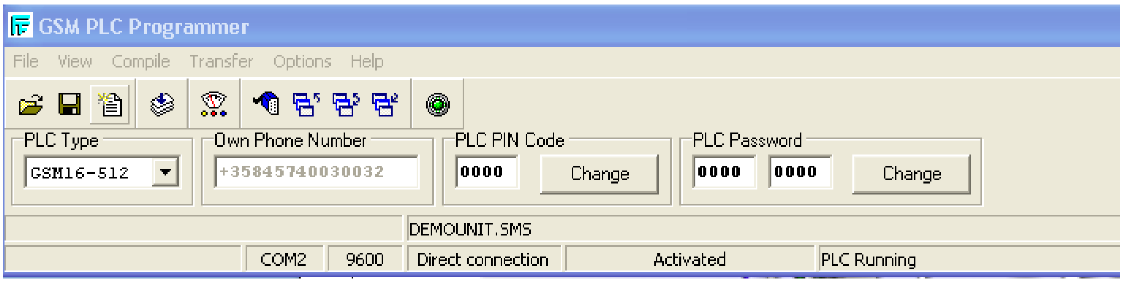
- Lists of *Phonebook* -page
- Lists and graphs of *Receiving_History* -page

11.4 Testing the FTP_File_Parser –driver, ftpdata –database and IWS_FTP_Parser_Demo

Load IWS GSM-demo application to GSM-PLC. This application collects FTP file and sends it to FTP server or GSMProgrammer's project directory. For 1st step testing, you don't need to have FTP server in your PC and you don't need to have SIM card in your GSM-PLC. You can use direct cable connection.



if the GSM-PLC cable is connected between PC and GSM-PLC and DIP switch 4 is set to "ON" position and you have open connection between GSM-Programmer and GSM-PLC



PLC Type: GSM16-512

Own Phone Number: +35845740030032

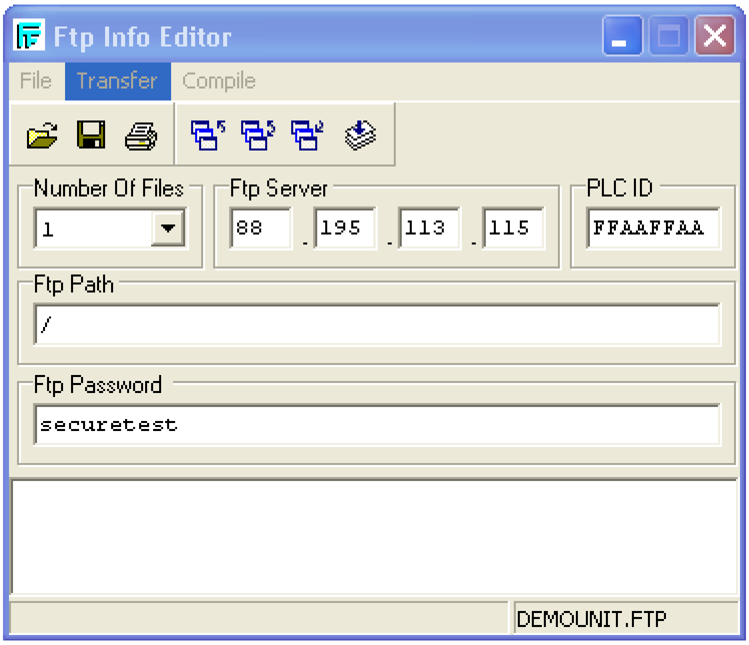
PLC PIN Code: 0000

PLC Password: 0000 0000

DEMOUNIT.SMS

COM2 | 9600 | Direct connection | Activated | PLC Running

It can take about 1 minute for GsmProgrammer to connect with GSM-PLC. Whe its connected you can see that the communication LEDs are blinking fast and the Status line text "halted" changes to "Activated".



Number Of Files: 1

Ftp Server: 88 . 195 . 113 . 115

PLC ID: FFAAFFAA

Ftp Path: /

Ftp Password: securetest

DEMOUNIT.FTP

Open Ftp Info Editor:
View->Ftp info

Change PLC ID to FFAAFFAA (or the same as your trend table name suffix earlier when you created the table for ftpdata database trend_ffaffaa

If you can check the PLC_ID also with Alarm Log –view
View->Alarm Log

FTP ? **;Returns FTP configurations**

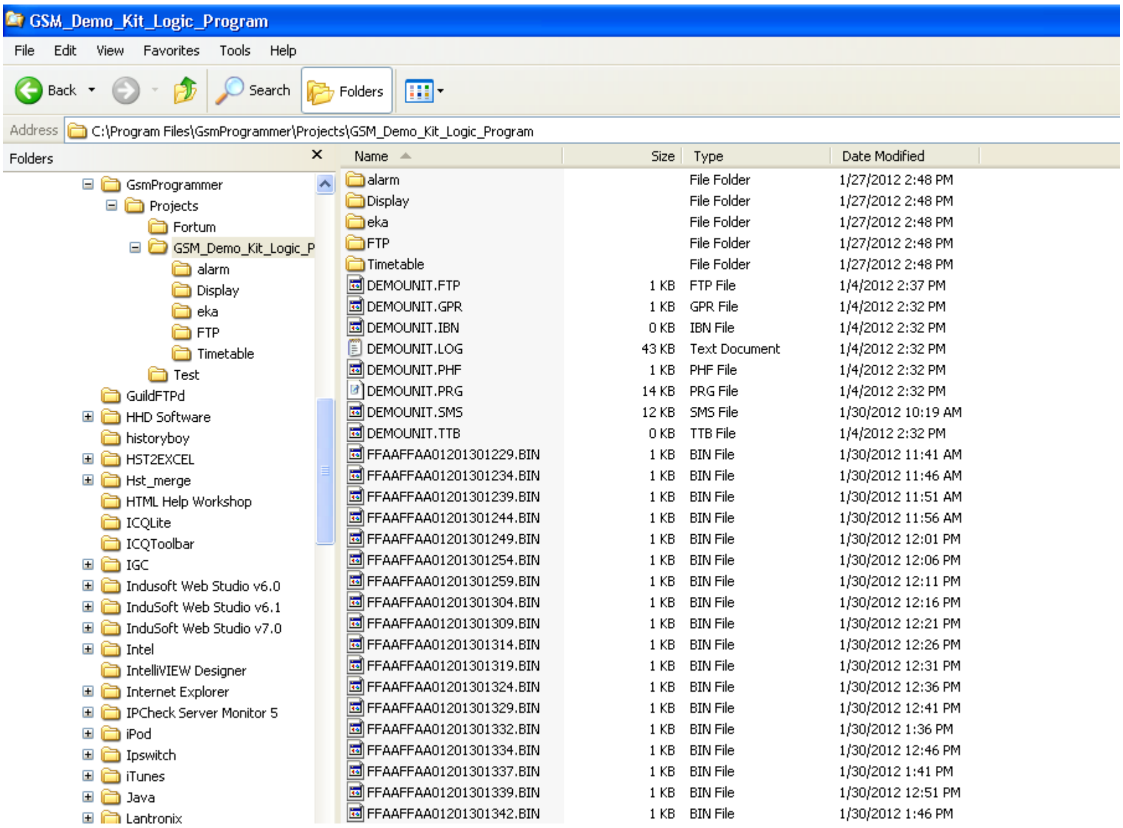
```
0000 FTP ?*
+35812345678, #FTP =1,222.118.20.68,/,FFAAFFAA,securetest
```

Alarm log also informs when it has received the FTP file from the GSM-PLC. These files are stored to the project directory.

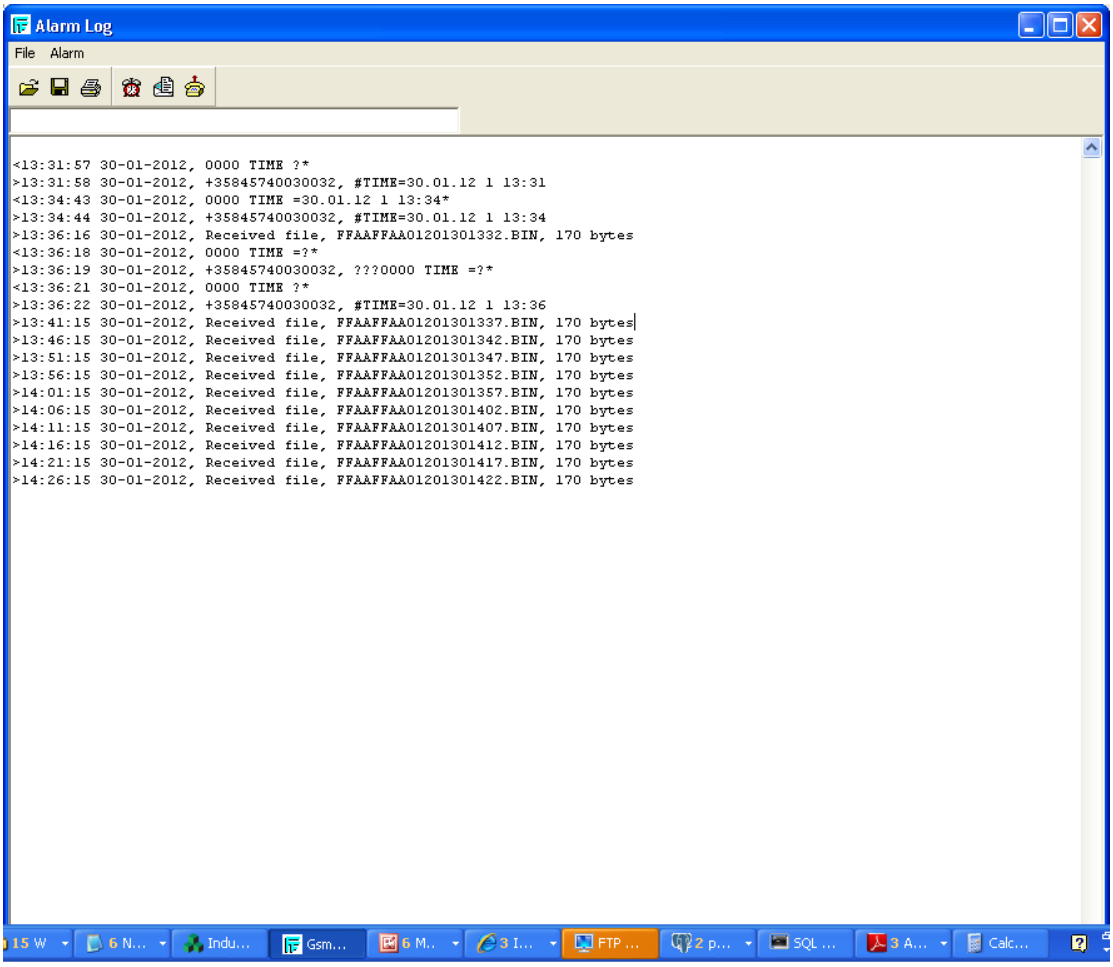
e.g.

C:\Program Files\GsmProgrammer\Projects\GSM_Demo_Kit_Logic_Program

Appendix



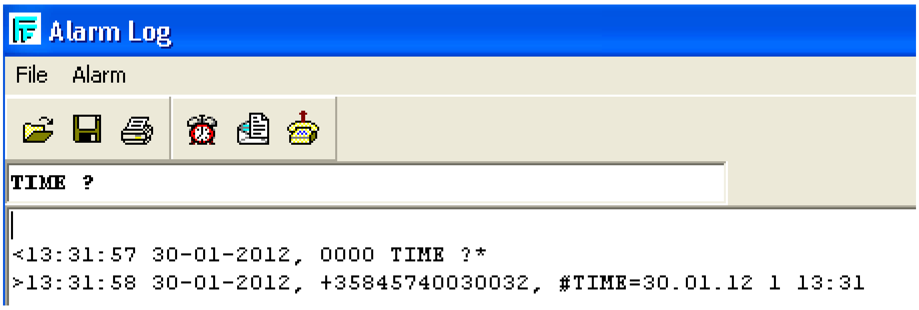
Name	Size	Type	Date Modified
alarm		File Folder	1/27/2012 2:48 PM
Display		File Folder	1/27/2012 2:48 PM
eka		File Folder	1/27/2012 2:48 PM
FTP		File Folder	1/27/2012 2:48 PM
Timetable		File Folder	1/27/2012 2:48 PM
DEMOUNIT.FTP	1 KB	FTP File	1/4/2012 2:37 PM
DEMOUNIT.GPR	1 KB	GPR File	1/4/2012 2:32 PM
DEMOUNIT.IBN	0 KB	IBN File	1/4/2012 2:32 PM
DEMOUNIT.LOG	43 KB	Text Document	1/4/2012 2:32 PM
DEMOUNIT.PHF	1 KB	PHF File	1/4/2012 2:32 PM
DEMOUNIT.PRG	14 KB	PRG File	1/4/2012 2:32 PM
DEMOUNIT.SMS	12 KB	SMS File	1/30/2012 10:19 AM
DEMOUNIT.TTB	0 KB	TTB File	1/4/2012 2:32 PM
FFAAFFAA01201301229.BIN	1 KB	BIN File	1/30/2012 11:41 AM
FFAAFFAA01201301234.BIN	1 KB	BIN File	1/30/2012 11:46 AM
FFAAFFAA01201301239.BIN	1 KB	BIN File	1/30/2012 11:51 AM
FFAAFFAA01201301244.BIN	1 KB	BIN File	1/30/2012 11:56 AM
FFAAFFAA01201301249.BIN	1 KB	BIN File	1/30/2012 12:01 PM
FFAAFFAA01201301254.BIN	1 KB	BIN File	1/30/2012 12:06 PM
FFAAFFAA01201301259.BIN	1 KB	BIN File	1/30/2012 12:11 PM
FFAAFFAA01201301304.BIN	1 KB	BIN File	1/30/2012 12:16 PM
FFAAFFAA01201301309.BIN	1 KB	BIN File	1/30/2012 12:21 PM
FFAAFFAA01201301314.BIN	1 KB	BIN File	1/30/2012 12:26 PM
FFAAFFAA01201301319.BIN	1 KB	BIN File	1/30/2012 12:31 PM
FFAAFFAA01201301324.BIN	1 KB	BIN File	1/30/2012 12:36 PM
FFAAFFAA01201301329.BIN	1 KB	BIN File	1/30/2012 12:41 PM
FFAAFFAA01201301332.BIN	1 KB	BIN File	1/30/2012 1:36 PM
FFAAFFAA01201301334.BIN	1 KB	BIN File	1/30/2012 12:46 PM
FFAAFFAA01201301337.BIN	1 KB	BIN File	1/30/2012 1:41 PM
FFAAFFAA01201301339.BIN	1 KB	BIN File	1/30/2012 12:51 PM
FFAAFFAA01201301342.BIN	1 KB	BIN File	1/30/2012 1:46 PM



```
<13:31:57 30-01-2012, 0000 TIME ?*
>13:31:58 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:31
<13:34:43 30-01-2012, 0000 TIME =30.01.12 1 13:34*
>13:34:44 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:34
>13:36:16 30-01-2012, Received file, FFAAFFAA01201301332.BIN, 170 bytes
<13:36:18 30-01-2012, 0000 TIME =?*
>13:36:19 30-01-2012, +35845740030032, ???0000 TIME =?*
<13:36:21 30-01-2012, 0000 TIME ?*
>13:36:22 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:36
>13:41:15 30-01-2012, Received file, FFAAFFAA01201301337.BIN, 170 bytes
>13:46:15 30-01-2012, Received file, FFAAFFAA01201301342.BIN, 170 bytes
>13:51:15 30-01-2012, Received file, FFAAFFAA01201301347.BIN, 170 bytes
>13:56:15 30-01-2012, Received file, FFAAFFAA01201301352.BIN, 170 bytes
>14:01:15 30-01-2012, Received file, FFAAFFAA01201301357.BIN, 170 bytes
>14:06:15 30-01-2012, Received file, FFAAFFAA01201301402.BIN, 170 bytes
>14:11:15 30-01-2012, Received file, FFAAFFAA01201301407.BIN, 170 bytes
>14:16:15 30-01-2012, Received file, FFAAFFAA01201301412.BIN, 170 bytes
>14:21:15 30-01-2012, Received file, FFAAFFAA01201301417.BIN, 170 bytes
>14:26:15 30-01-2012, Received file, FFAAFFAA01201301422.BIN, 170 bytes
```

11.4.1 Setting the PLC clock with Alarm Log

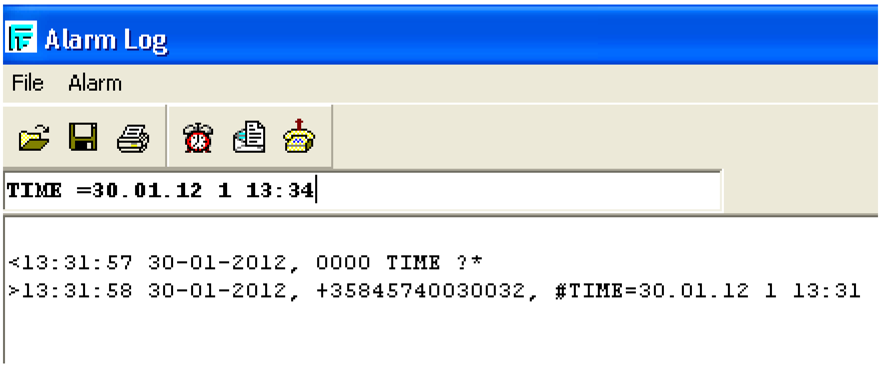
You can check and set the PLC clock to correct time.



```
TIME ?
```

```
<13:31:57 30-01-2012, 0000 TIME ?*
>13:31:58 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:31
```

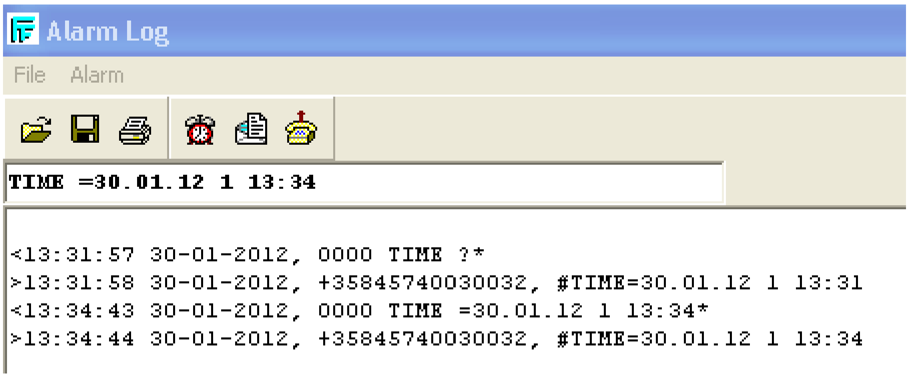
TIME ? ;returns the date and time, you can copy the time format for next command.



```
TIME =30.01.12 1 13:34
```

```
<13:31:57 30-01-2012, 0000 TIME ?*
>13:31:58 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:31
```

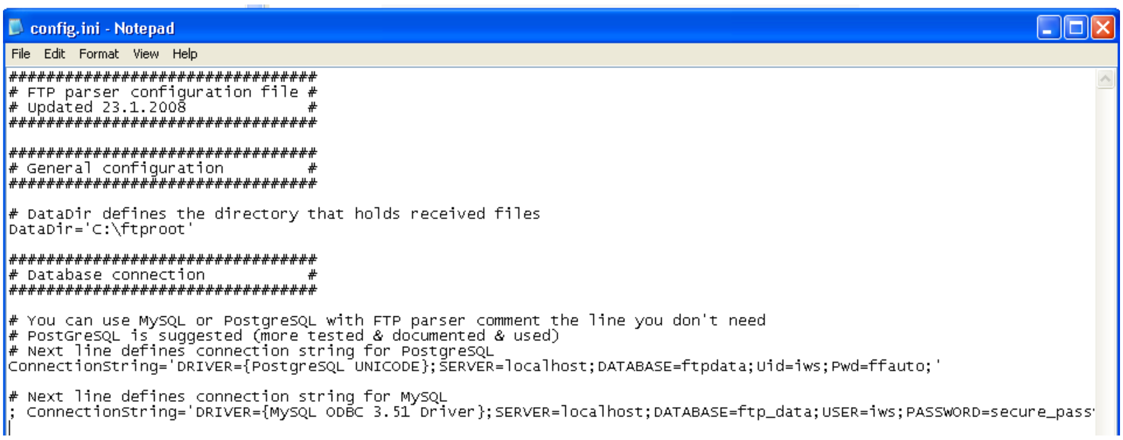
Paste the time format and add one space between TIME and "=", and change the time.



```
TIME =30.01.12 1 13:34
```

```
<13:31:57 30-01-2012, 0000 TIME ?*
>13:31:58 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:31
<13:34:43 30-01-2012, 0000 TIME =30.01.12 1 13:34*
>13:34:44 30-01-2012, +35845740030032, #TIME=30.01.12 1 13:34
```

Appendix

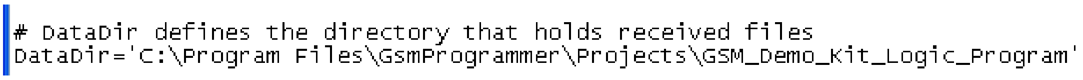


```
#################################
# FTP parser configuration file #
# Updated 23.1.2008             #
#################################

#################################
# General configuration         #
#################################

# DataDir defines the directory that holds received files
DataDir='C:\ftproot'

#################################
# Database connection           #
#################################

# You can use MySQL or PostgreSQL with FTP parser comment the line you don't need
# PostGreSQL is suggested (more tested & documented & used)
# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'

# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=ftp_data;USER=iws;PASSWORD=secure_pass'
```

Open ftpparser program’s config.ini

You can change the DataDir –line:

DataDir='C:\ProgramFiles\GsmProgrammer\Projects\GSM_Demo_Kit_Logic_Program'



```
# DataDir defines the directory that holds received files
DataDir='C:\Program Files\GsmProgrammer\Projects\GSM_Demo_Kit_Logic_Program'
```

ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=yourpostgresqlpassword;'

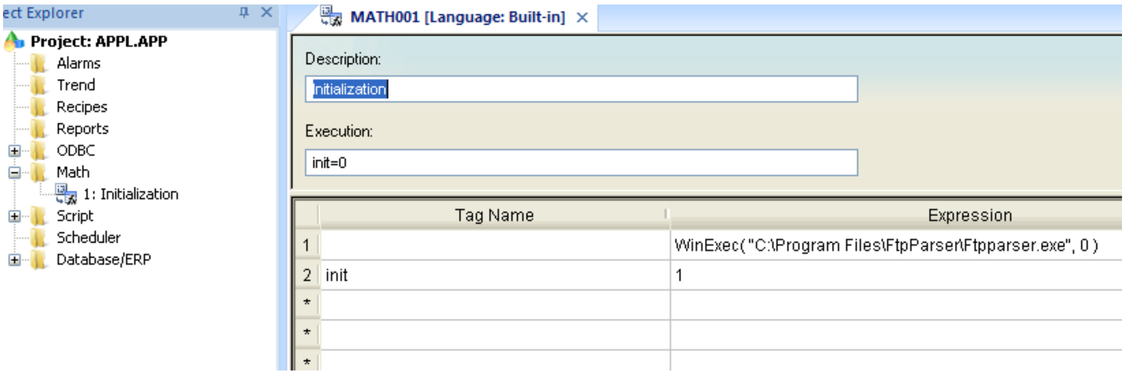
Save the file and double click FTPParser.exe

40 KB FTPParser.exe Application 1/24/2008 4:26 PM

FTPParser starts in the system tray. You can exit the program by double right clicking the icon.

8 KB state_log.txt Text Document 1/30/2012 2:47 PM

You can check the status of the FTPParser from the state_log.txt file.



Description: Initialization

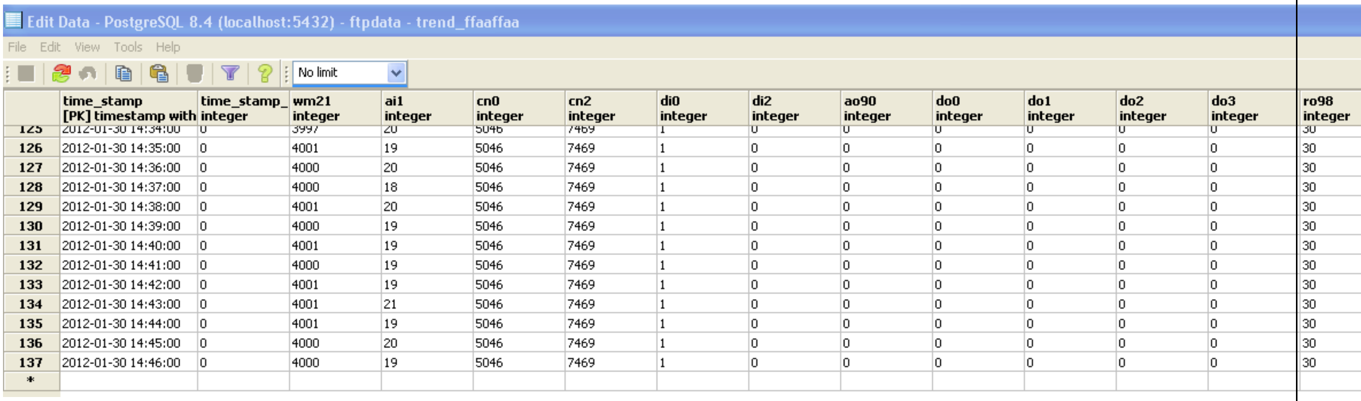
Execution: init=0

	Tag Name	Expression
1		WinExec("C:\Program Files\FtpParser\Ftpparser.exe", 0)
2	init	1
*		
*		
*		

FTP parser can be started automatically from Indusoft using the following math –task.

WinExec("C:\Program Files\FtpParser\Ftpparser.exe", 0)

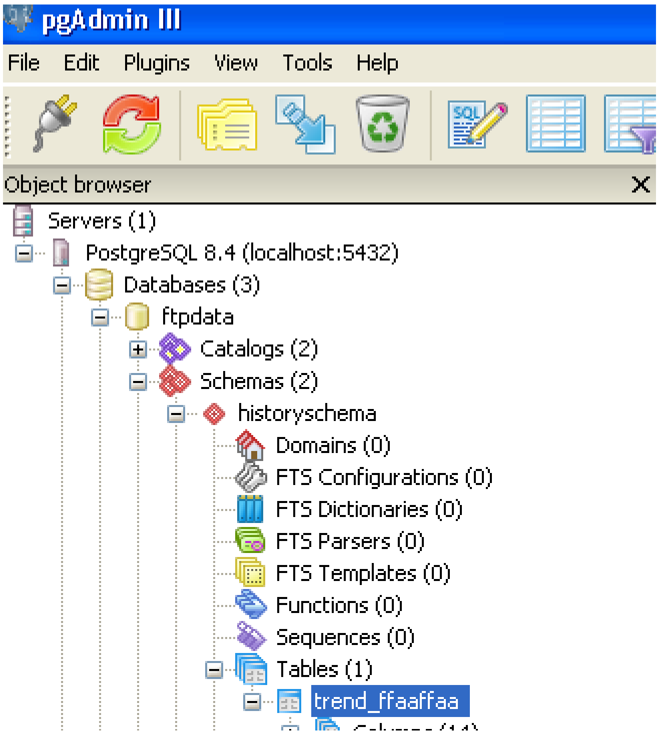
Appendix



Edit Data - PostgreSQL 8.4 (localhost:5432) - ftpdata - trend_ffaaffaa
File
Edit
View
Tools
Help
No limit

	time_stamp [PK] timestamp with	time_stamp_ integer	wm21 integer	ai1 integer	cn0 integer	cn2 integer	di0 integer	di2 integer	ao90 integer	do0 integer	do1 integer	do2 integer	do3 integer	ro98 integer
125	2012-01-30 14:34:00	0	3997	20	5046	7469	1	0	0	0	0	0	0	30
126	2012-01-30 14:35:00	0	4001	19	5046	7469	1	0	0	0	0	0	0	30
127	2012-01-30 14:36:00	0	4000	20	5046	7469	1	0	0	0	0	0	0	30
128	2012-01-30 14:37:00	0	4000	18	5046	7469	1	0	0	0	0	0	0	30
129	2012-01-30 14:38:00	0	4001	20	5046	7469	1	0	0	0	0	0	0	30
130	2012-01-30 14:39:00	0	4000	19	5046	7469	1	0	0	0	0	0	0	30
131	2012-01-30 14:40:00	0	4001	19	5046	7469	1	0	0	0	0	0	0	30
132	2012-01-30 14:41:00	0	4000	19	5046	7469	1	0	0	0	0	0	0	30
133	2012-01-30 14:42:00	0	4001	19	5046	7469	1	0	0	0	0	0	0	30
134	2012-01-30 14:43:00	0	4001	21	5046	7469	1	0	0	0	0	0	0	30
135	2012-01-30 14:44:00	0	4001	19	5046	7469	1	0	0	0	0	0	0	30
136	2012-01-30 14:45:00	0	4000	20	5046	7469	1	0	0	0	0	0	0	30
137	2012-01-30 14:46:00	0	4000	19	5046	7469	1	0	0	0	0	0	0	30
*														

You can check that the data is coming to database from pgAdminIII –program.

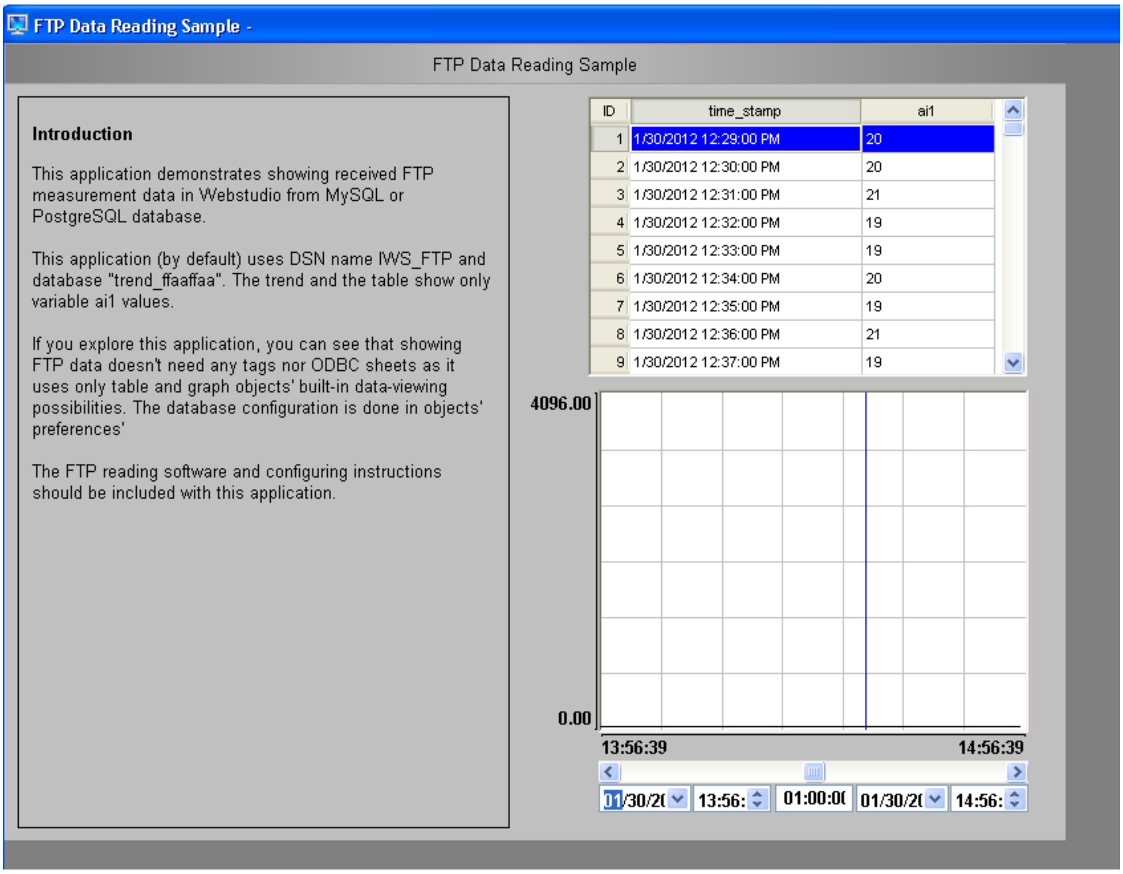


Right click and select ->View data->view all

Now open Indusoft web studio IWS_FTPParserDemo.app

Check that you have set correct DSN name (IWS_FTP) in the screen objects.

RUN the project.



Introduction

This application demonstrates showing received FTP measurement data in Webstudio from MySQL or PostgreSQL database.

This application (by default) uses DSN name IWS_FTP and database "trend_ffaaffaa". The trend and the table show only variable ai1 values.

If you explore this application, you can see that showing FTP data doesn't need any tags nor ODBC sheets as it uses only table and graph objects' built-in data-viewing possibilities. The database configuration is done in objects' preferences'

The FTP reading software and configuring instructions should be included with this application.

ID	time_stamp	ai1
1	1/30/2012 12:29:00 PM	20
2	1/30/2012 12:30:00 PM	20
3	1/30/2012 12:31:00 PM	21
4	1/30/2012 12:32:00 PM	19
5	1/30/2012 12:33:00 PM	19
6	1/30/2012 12:34:00 PM	20
7	1/30/2012 12:35:00 PM	19
8	1/30/2012 12:36:00 PM	21
9	1/30/2012 12:37:00 PM	19

You should receive data to the tables.

This is just a simple demo. You can also configure ODBC task which reads values from the database to Indusoft tags.

12 Appendix

12.1 DB2SMS –driver interface description

DB2SMS is SMS –driver, which handles the outgoing and incoming SMS messages. It is responsible for “low level” communication with GSM modem, filtering the incoming SMS messages and “parsing” the messages. DB2SMS uses an PosGreSQL –database interface with other programs (like Indusoft Web Studio (IWS) SCADA software). DB2SMS driver can receive and send SMS messages through GSM modem with FF-Automation’s AutoLog GSM-RTU devices.

DB2SMS supports three kinds of incoming (GSM-RTU->IWS) SMS message types:

1. *Report* –message can be used to send the latest values of GSM-RTU’s variables/Inputs. One message can have many variables. One value / variable.
2. *History* –message can be used to send many time-stamped values of one variable.
3. *Event* –messages can be used to send other kinds of text messages e.g. alarms or events.

DB2SMS supports three kinds of outgoing (IWS->GSM-RTU) SMS message types:

1. Send Group SMS –message (Same message is send to all defined GSM numbers)
2. Send one SMS –message (Message is send to one defined GSM number)
3. send System Error SMS -message. If the DB2SMS detects that IWS application is not running (watchdog pulse from IWS is not changing), it can send System Error SMS –message.

DB2SMS is supplied with DEMO project’s DB2SMS -sub folder.

12.1.1 DB2SMS version history

v.1.0

- 5.1.2007 (ArMe)
- Initialization of GSM modem
- Sending SMS messages (to one, to group, fault events messages)
- Receiving SMS messages (report, history and event message types)
- Program status in database
- Compatible with different databases

12.1.2 Needed components etc.

- Indusoft Web Studio SCADA (Or some other SCADA with ODBC support*)
- DB2SMS program
- Win2000/XP OS
- Microsoft .Net Framework 2.0 (can be downloaded from Microsoft homepage)
- GSM-modem
- Database, which support ODBC –interface (MySQL, PostgreSQL)
- Note! Indusoft web studio demo uses PostgreSQL Database connection.

* Please contact FF-Automation for more information!

12.1.3 Installing DB2SMS

See separate installation chapter!

12.1.4 Configuring DB2SMS Program

12.1.5 General configuration description

When configuring the program you should check the following parameters:

- PIN code
- The serial port which is used by GSM modem
- The location of the database, DNS name, user name and password

8. Open *config.ini* with text editor. The file looks like this:

```
####################################
# Modem settings
####################################

Modem_Port='1'
#Modem_PIN='0000'
Modem_Speed='9600'

# (0 = No handshaking, 1 = XOnXOff, 2 = RequestToSend, 3 = RequestToSendXOnXOff)
; Modem_Handshaking='2'
Modem_Handshaking='1'

####################################
# Database configuration
#
# See http://www.connectionstring.com
# if you don't want use PostgreSQL or
# MySQL
####################################

# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51
;Driver};SERVER=localhost;DATABASE=smsindusoft;USER=root;PASSWORD=securepassword;'

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=smsindusoft;Uid=iws;Pwd=ffauto;'

####################################
# Wierd options
# These may be useful in some cases
########################################

# Next line defines how many times modem initialization may fail before error message is given
# We've noticed that a modem gives "modem link is too weak" -alarm without any
# reason in some cases. Therefore we recommend not to comment next line.

WT_InitializationErrorMax='2'
```

(Same config.ini is also in the appendix)

9. Used serial port for modem is defined in the first line.
10. SIM –card's PIN number is defined in the second line. You can leave this undefined if the *PIN code request* is configured as 'off' in your SIM card.
11. Modem speed is defined in the third line. Normally this is 9600, so normally you do not need to change it.
12. Modem_Handshaking parameter is important and it should be defined correctly. If you are unsure which value to use, you can try one by one which work correctly. Look the table next table for more information!
13. The connection string defines the database connection. MySQL connection string is commented. Change the user name and password to correct ones.

- DB2SMS uses ODBC –driver to create connection with database.
- MySQL
  - Uses MyODBC database connection driver. This can be downloaded for example:
  - http://dev.mysql.com/downloads/connector/odbc/3.51.html
  - ConnectionString should include Server, Database, User and Password –parameters (look example file)
- PostgreSQL
  - PostgreSQL includes needed ODBC connection drivers. You don't need to install these separately.
  - ConnectionString should include Server, Database, User and Password –parameters (look example file)
- Access
  - Access 2000 –database is not supported. If you need to use this, you should use Access2SMS program, not DB2SMS program. Note that DB2SMS program has more features than Access2SMS.
- Other databases

- Other databases are not tested with DB2SMS. In principle all the databases that uses ODBC interface should work without problems.

14. Normally you don't need to give any other parameters to DB2SMS, but in the next table includes all the parameters that can be configured.

Parameter	Mandatoryn/Optional	Description
ConnectionString='x'	Mandatory	Database connection definition
Modem_Port='x'	Mandatory	Used serial port for GSM modem
Modem_Speed	Mandatory	Serial port speed (normally 9600)
Modem_Handshaking	Mandatory	Handshaking level 0 = No handshaking 1 = XOn / XOff 2 = Request to send 3 = Request to send and XOn / XOff
pin='xxxx'	Optional	SIM –card's PIN –code
license='move'	Optional	Activates licence move state
loglevel='x'	Optional (default 1)	Selects the level for message log: 1:Received and send messages are written to log (default). 2:The serial communication is written to log. 3: Both 1 and 2.

Config.ini Parameter explanations

Note! If your PIN code request is enabled in your SIM card, you should not change the PIN code while the program is running. DB2SMS tries to initialize SIM card with the defined PIN code, and if the code is wrong, the SIM card probably gets locked.

12.1.6 Troubleshooting - Modem initialization problems

If you have problems with initializing modem you can check that these are ok:

- You have inserted SIM card
- PIN code request is disabled / you have right PIN code in config.ini
- You have right type of cable and it is in right COM port

- Baudrate is set correctly (9600)
- Try different options 1-3 for Modem_Handshaking='1' parameter.

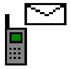
If the above were ok you can test that the modem is in TEXT mode and the ECHO is OFF.

Open HyperTerminal and start session with parameters COM#, 9600,8/1/N,OFF

Command	Modem should Reply
AT	OK
AT+CPIN?	OK or +CPIN: SIM PIN or +CPIN: READY If you get ERROR or +CPIN: SIM PUK, there's a Problem with the SIM.
AT+CMGF?	1 (=text mode)
ATE0	OK (sets echo off)

12.1.7 Starting the DB2SMS program

Program can be started by double-clicking the *DB2SMS.exe* file. If your program is unlicensed, the program informs how much evaluation time you have left and when clicking the *Continue Evaluate software* –button the application is started. If the configuration of the program is correct you should see small "mobile phone" picture in the right down corner (system tray):



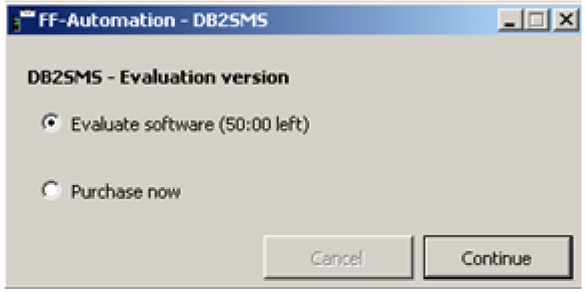
12.1.8 Closing the program

Program can be closed by double clicking the mouse's **right** button on the "mobile phone" picture in the right down corner (system tray). After that you will get confirmation message and answering 'Yes' closes the program.

12.1.9 Program license

12.1.9.1 Evaluation period

DB2SMS program can be used 50 hours evaluation time without licensing. Program inform how much time is left when it is started.



FF-Automation - DB2SMS

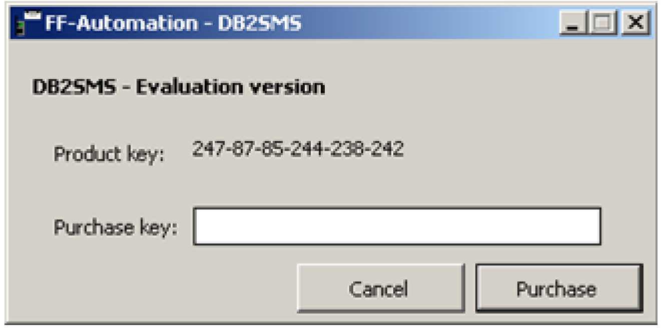
DB2SMS - Evaluation version

● Evaluate software (50:00 left)

○ Purchase now

12.1.9.2 Purchase and registration

Program can be licensed by selecting the *Purchase now* checkbox and clicking *Continue*. Program generates product key, which is unique key for PC. This key should then be send to FF-Automation (e.g. by e-mail) and FF-Automation then sends back the right *Purchase key* to activate the license. Program informs if the activation were successful.



FF-Automation - DB2SMS

DB2SMS - Evaluation version

Product key: 247-87-85-244-238-242

Purchase key:

12.1.9.3 Moving the license

The license can be moved to other PC. The original license is removed.

You should first install DB2SMS to destination PC. After that you should add the following line to the original already licensed PC's configuration file (Config.ini)

license='move'

Look the previous Config .ini parameter explanations –table!
This activates the license transfer state. Write the Product key –field the destination PC's Product key and click Move. Program warns about the license transfer, if you answer 'Yes' the original license is removed and you get new Purchase key to be inserted to destination PC's Purchase key –field.

Note! Be careful when you insert the product key. If the product key is faulty, the program doesn't generate right purchase key and you will lose the license. If this happens please contact with FF-Automation. (Include the product key and the generated purchase key in your message.)

12.1.10 Receiving SMS messages

Program can receive three types of messages: report-, history data- and event messages.

The program goes through the *Phones_Receiving* table in the database. Only the messages which are coming from the active phone numbers (enabled <>0) are accepted.

12.1.11 Event message (*Received_Events*)

Event message can be used to send e.g. alarm, warning, status change or any other freely defined message from the GSM-PLC. Message consists of Message type identifier (A), Event type (nn as integer), DateTime (DDMMYYHHMMSS) and Actual message.

12.1.11.1 Format

Ann,DDMMYYHHMMSS,Message

12.1.11.2 Examples

Receiving event message "Problem" with date and time (ID 0):

A0,060705150000,Problem

Receiving event message "Problem" without date and time (ID 1):

A1,,Problem

Received events are stored in Received_Events –table. Look more from database structure chapter!

12.1.12 History (Received_History)

History message can be used to send one variables history e.g. some sensor's measurement value history. History message uses Variable_Crosslinks –table.
Message consists of Message type identifier (H), variable number (nn as integer), DateTime (DDMMHHYYMMSS), Sample Interval (HHMM) and actual data (D1,D2…)
The data part starts from the most recent value (D1), the oldest value comes last.

12.1.12.1 Format

Hnn,DDMMYYHHMMSS,HHMM,D1,D2,D3,D4…

12.1.12.2 Examples

Receiving history message from variable number 1, Sample interval 1 hour with DateTime:

H1,060705150000,0100,20,19,19,18,18,19,17,15,14,13,12,12

Receiving history message from variable number 2, Sample interval 3 hour 30 min without DateTime:

H2,0230,20,19,19,18,18,19,17,15,14,13,12,12

Received history messages are stored in Received_History –table. The data is divided in separate rows in the table. The variable number is changed to variable name using Variable_Crosslinks -table. Look more from database structure chapter!

12.1.13 Report (Received_Report)

Report message can be used to send the latest values of many variables. Report message consists of Message type identified (D), first variable number (nn as integer), DateTime (DDMMHHYYMMSS) and actual data (Variable values).

Program uses *Variable_Crosslink* –table to change the variable number to variable name. Every variable in the messages data part increases variable's number by one so that first variable's V1 number is nn, V2 is nn+1…

12.1.13.1 Format

D*nn,DDMMYYHHMMSS,V1,V2,V3,V3…*

12.1.13.2 Examples

Receiving Report message from variable values numbers 0,1 and 2 with DateTime:

D0,060705150000,20,19,21

Receiving Report message from variable values numbers 1,2 and 3 without DateTime:

D1,,20,19,21

12.1.14 Sending message(s)

Message sending can be done in two ways: To one phone number (Send to one) or to Group of phone numbers (Send to group).

12.1.15 Send to Group messages

Program sends all messages in the *SendAll* table which has flag = 1 value to phone numbers defined in *Phones_Sending* table which are active (enabled =1).

12.1.16 Send to One messages

Program sends all messages in the *Send_One* table which has flag =1. Before sending the message user should write value to *Phone* –field and after that change the flag to 1 which sends the message.

12.1.17 Error messages

Server program needs to send e.g. increasing counter value to *PRG_Counter* –field. If the program detects that this value has not changed and the time in *A2SMS_Counte*r has increased to *Counter_Max* it sends defined message to phone number and put flag to zero. When the value in the *PRG_Counte*r –field changes it changed flag to 1.

12.1.18 Log files

There are three log files which can be used to debug the possible fault situations etc. The logging level is configured in Config.ini file.

File name	description

sms_traffic.txt serial_traffic.txt state_log.txt	SMS traffic Serial traffic Program states (Error codes)

12.1.19 Error Codes

In addition to state_log, DB2SMS stores its states also to database (*Errormessage* -table).

Error code	Description
0	No errors
Modem Errors (write to log and database)	
M01x	Could not open serial port
M02x	Message couldn’t be send to serial port
M03x	Modem is not answering
M04x	PIN code error (wrong PIN / SIM locked)
M05x	Could not go to Text mode
M06y, M07x	GSM modem signal strenght information sending disabling failed
M08x	Unknown problem when initialization modem
M1x	Error in receiving message
M2x	Error in sending message
M3x	Error in calculating new messages
Critical Errors (write to log and database (if possible), info to screen and stops the program)	
C00	Critical program error (Unknown error – report to FF-Automation!)
C02	PIN code insertion error
C03	Database not found
Errors in message handling (write to log only)	
P0	DateTime format incorrect
P1	Report message format incorrect
P2	Message format incorrect
P3	Error in saving message

12.1.20 Database stucture

- **ErrorMessage (Programs writes)**
  - Errorcode (String)
    - Look Error codes chapter!
- **Phonebook_Sending (User defines)**
  - Name (String)
    - Program doesn’t use this field. It can be used in SCADA program.
  - Phone (String)
    - Phone number
  - Enabled (Integer)
    - 0: Phone number not in use
    - 1: Phone number in use
- **Phonebook_Receiving (User defines)**
  - Name (String)
    - Program reads the field to Received tables. Field can be used also in SCADA program.
  - Phone (String)
    - Phone number (Needs land code e.g. +358401234)
  - Enabled
    - 0: Phone number not in use
    - 1: Phone number in use
- **Received_Events Program writes)**
  - Mindex (Integer, increasing)
    - Unique identifier for every received message
  - Flag (Integer)
    - Value 1 indicates that the message is new. The user program (e.g. SCADA) can reset this value to zero to indicate that it has read the message. Old messages can be deleted e.g. after reading /once per day or week to protect that the table size doesn’t increase too much.
  - Date (String)
    - Message receiving date (yyyy-mm-dd)
  - Time (String)
    - Message receiving time (hh:mm:ss)
  - Phone (String)
    - Phone number from where the message was received

- Name (String)
    - Sender’s name (copied from *Phonebook_Receiving* -table)
- Event (String)
    - Actual message, ”free format”
- ID (Integer)
    - Event ID

- **Received_Report (Program writes)**
    - Mindex (Integer, increasing)
        - Unique identifier for every received message
    - Flag (Integer)
        - Value 1 indicates that the message is new. The user program (e.g. SCADA) can reset this value to zero to indicate that it has read the message. Old messages can be deleted e.g. after reading /once per day or week to protect that the table size doesn’t increase too much.
    - Date (String)
        - Message receiving date (yyyy-mm-dd)
    - Time (String)
        - Message receiving time (hh:mm:ss)
    - Phone (String)
        - Phone number from where the message was received
    - Name (String)
        - Sender’s name (copied from *Phonebook_Receiving* -table)
    - Variable (String)
        - Variable’s name. DB2SMS copies this name from *Report_Crosslinks* –table’s name field using the variable number.
    - Value (String)
        - Variable’s value
        - String type. Supports both text and number formats. This allows variable’s value to be send also in text format.
- **Received_History**
    - Mindex (Integer, increasing)
        - Unique identifier for every received message
    - Flag (Integer)

        - Value 1 indicates that the message is new. The user program (e.g. SCADA) can reset this value to zero to indicate that it has read the message. Old messages can be deleted e.g. after reading /once per day or week to protect that the table size doesn't increase too much.
    - Date (String)
        - Message receiving date (yyyy-mm-dd)
    - Time (String)
        - Message receiving time (hh:mm:ss)
    - Phone (String)
        - Phone number from where the message was received
    - Name (String)
        - Sender's name (copied from *Phonebook_Receiving* -table)
    - Variable (String)
        - Variable's name (Read from the message)
    - Value (String)
        - Variable's value
- **Send_All (User defines, program changes)**
    - Mindex (Integer, increasing)
        - Unique identifier for every send message
    - Flag (Integer)
        - Send message –flag. This is reset to zero after sending the message.
    - Message (String)
        - Send message
        - 160 characters
- **Send_One (User defines, program changes)**
    - Mindex (Integer, increasing)
        - Unique identifier for every send message
    - Flag (Integer)
        - Send message –flag. This is reset to zero after sending the message.
    - Phone (String)
        - Phone number
    - Message (String)
        - Send message

        - 160 characters
- **Variable_Crosslinks (User defines)**
    - ▪ Index (Integer)
        - Variables index number
    - ▪ Name (String)
        - Variable's name
        - Note that e.g. Web Studio SCADA doesn't support special characters or spaces in this field!
- **Send_Watchdog (User defines partly, program changes)**
    - ▪ Name (String)
        - Identifier name field for e.g. SCADA application use. Not used by DB2SMS.
    - ▪ Enabled (Integer)
        - Watchdog enabled when 1, user defines.
    - ▪ Flag (Integer)
        - Changes to 1, when PRG_Counter and Old_PRG_Counter has different values for too long time.
        - (1=proram ok, 0=problem in program message send)
    - ▪ PRG_Counter (Integer)
        - SCADA should change this using e.g. increasing counter.
    - ▪ Old_PRG_Counter (Integer)
        - ~10 sec intervals the value from PRG_Counter is copied to this
    - ▪ A2SMS_Counter (Integer)
        - If PRG_Counter and Old_PRG_Counter has same values, the value will be increased by one (~10 seconds intervals). If they differ this value is reset to zero.
    - ▪ Counter_Max (Integer)
        - Max value for A2SMS_Counter before sending the message and flag reset to zero.
    - ▪ Phone (String)
        - Phone number to where the warning message is send
    - ▪ Message (String)
        - Send warning message ”free format”

12.2 DB2SMS Config.ini example

```
########################################
# Modem settings
########################################

Modem_Port='1'
#Modem_PIN='0000'
Modem_Speed='9600'

# (0 = No handshaking, 1 = XOnXOff, 2 = RequestToSend, 3 = RequestToSendXOnXOff)
; Modem_Handshaking='2'
Modem_Handshaking='1'

########################################
# Database configuration
#
# See http://www.connectionstring.com
# if you don't want use PostgreSQL or
# MySQL
########################################

# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51
Driver};SERVER=localhost;DATABASE=smsindusoft;USER=root;PASSWORD=securepassword;'

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=smsindusoft;Uid=iws;Pwd=ffauto;'

########################################
# Wierd options
# These may be useful in some cases
######################################

# Next line defines how many times modem initialization may fail before error message is given
# We've noticed that a modem gives "modem link is too weak" -alarm without any
# reason in some cases. Therefore we recommend not to comment next line.

WT_InitializationErrorMax='2'
```

12.3 FTP_File_Parser Description

12.3.1 FTP History File Parser Version history

v.1.0

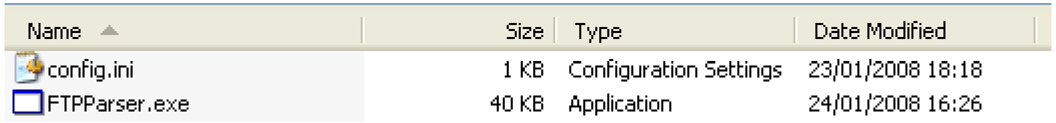
- 5.1.2007 (ArMe)
- Software copies history data from received FTP files into a database

12.3.2 Installation

12.3.3 Installing program file

Installation of FTP History File Parser is quite a simple process:

5. Install Microsoft .NET Framework 2.0 – not needed for Windows 7
6. Install PostgreSQL (see previous chapter)
7. Copy FTPPaerser.exe program executable and *config.ini* into local computer
8. Do necessary modifications to *config.ini* -file
9. Installation is ready



Name	Size	Type	Date Modified
config.ini	1 KB	Configuration Settings	23/01/2008 18:18
FTPParser.exe	40 KB	Application	24/01/2008 16:26

The application can be started simply by double clicking program icon. It'll run background until you close it by double clicking the program icon with the right button of your mouse in System tray.

```
###########################################
# FTP parser configuration file #
# Updated 23.1.2008             #
###########################################

###########################################
# General configuration         #
###########################################

# DataDir defines the directory that holds received files
DataDir='C:\FTP'

###########################################
# Database connection           #
###########################################

# You can use MySQL or PostgreSQL with FTP parser comment the line you don't need

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL ANSI};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'
;ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'

#for 64 bit windows 7
;ConnectionString='DRIVER={PostgreSQL Unicode(x64)};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'


# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51
Driver};SERVER=localhost;DATABASE=ftp_data;USER=iws;PASSWORD=secure_password;'
```

12.3.4 File format

A PLC sends a binary file, which always has two sections: PLC description section and data section.

12.3.5 PLC description block

PLC description section includes information about PLC and its variable configuration.

12.3.6 PLC description section format

Bytes 0-3 : PLC ID
Bytes 4-5 : Variable count
Bytes 6-> : Variable descriptors

Variable descriptors consist from two bytes. The first byte defines the variable type and the second byte defines the id of the variable.

E.g. If the first variable has value 5 and the second value 230, then the variable is RO230.

ID	Name
0	DI
1	DO
2	M
3	AI
4	CN
5	RO
6	WM
7	AO

12.3.7 Variable data block

After PLC description block exists data block. The format is quite a simple:
Bytes 0-3 : Date and time
Byte 4-> : Variable values (two bytes per variable)

The timestamp is encoded so that it need only four bytes.

First byte's seven most significant bits represents the value of year. The least significant bit of first byte and four most significant bits of second byte represents the value of month. Fife least significant bits of second byte represents the day. Third byte represents hour and fourth byte represents minute.

12.3.8 Date

1. byte	2. byte
YYYYYYYM	MMMDDDDD

12.3.9 Time

3. byte	4. byte
HHHHHHHH	MMMMMMMM

Variable values are located after timestamp. The variable sort should be same as in the PLC description block.

12.3.10 Example file

FF AA FF AA 00 01 07 00 0E 21 09 00 FF FF

- First four bytes represents the id, so the PLC id is FFAAFFAA
- The file has only one variable (00 01) and it is AI0 (07 00).
- The first (and the only) data field has date 1.1.2007 (0E 21) and time 09:00 (09 00).
- The variable AO0 value is 65535 (FF FF).

12.4 FTP_parser Config.ini example

```
####################################
# FTP parser configuration file #
# Updated 23.1.2008             #
####################################

####################################
# General configuration         #
####################################

# DataDir defines the directory that holds received files
DataDir='C:\FTP'

####################################
# Database connection           #
####################################

# You can use MySQL or PostgreSQL with FTP parser comment the line you don't need

# Next line defines connection string for PostgreSQL
ConnectionString='DRIVER={PostgreSQL ANSI};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'
#for 64 bit windows 7
;ConnectionString='DRIVER={PostgreSQL Unicode(x64)};SERVER=localhost;DATABASE=ftpdata;Uid=iws;Pwd=ffauto;'


# Next line defines connection string for MySQL
; ConnectionString='DRIVER={MySQL ODBC 3.51
Driver};SERVER=localhost;DATABASE=ftp_data;USER=iws;PASSWORD=secure_password;'
```

GSM-PLC application example: FTP pressure sending

```
;Program: FTP program
;Version: 1.1
;Date   : 23.5.2013
;Author : Antti Moijanen v.1.1 / (Jouni Paavonen v1.0 base code)

;*********I/O
;AI0=pressure1 (4...20mA I) (for 0-10bar sensor)
;AI1=pressure2(4...20mA I) (for 0-10bar sensor)
;AI2=temperature1 (-50...150)
;AI3=temperature2 (-50...150)

;********* FTP LOG & SEND PARAMETERS **********
;message in format "$FTP=5=60", where 5 defines log interval in minutes and 60 defiens the tx interval (in minutes)
;'($FTP)' M0 WM35=WM0 WM36=WM1
;'M0' "Log interval= %WM35 seconds, Tx interval = %WM36 seconds" 254
;Check for incorrect values

;'M100#!M100' WM35=30 ;seconds
;'WM35=0' WM35=1
;'WM36=0' WM36=5


'M100#!M100' WM22=AI2 WM23=AI3;executed always, moves RAW Analog input values to WM20-23

;********* FTP LOGGING & Transfer ***********

;If output status changes, log & send
;'DO0=1#DO0=0#DO1=1#DO1=0#DO2=1#DO3=1#DO3=0' M62=1              ;log & send on event

; Calculating 10 sek average

'P1' WM59=WM58 WM58=WM57 WM57=WM56 WM56=WM55 WM55=WM54 WM54=WM53 WM53=WM52 WM52=WM51 WM51=WM50 WM50=AI0
'P1' WM10=WM59 WM10+WM58 WM10+WM57 WM10+WM56 WM10+WM55 WM10+WM54 WM10+WM53 WM10+WM52 WM10+WM51 WM10+WM50 WM10/10
WM20=WM10
```

```
'P1' WM69=WM68 WM68=WM67 WM67=WM66 WM66=WM65 WM65=WM64 WM64=WM63 WM63=WM62 WM62=WM61 WM61=WM60 WM60=AI1
'P1' WM11=WM69 WM11+WM68 WM11+WM67 WM11+WM66 WM11+WM65 WM11+WM64 WM11+WM63 WM11+WM62 WM11+WM61 WM11+WM60 WM11/10
WM21=WM11

'P1' WM30+1             ;increment logging timer and transfer timer once/second (P1)
'P1' WM100+1                    ;increment communication watchdog once/second (P1)
'WM100=65535' WM100=1000        ;resets communication watchdog to 1000 when reached 65535

;**** LOGGING ****
'WM30=55' M60                           ;timer baser logging (M60)
'M60=1' "%WM20,%WM21,%WM22,%WM23,%WM100,%RO98" 245

;**** TRANSFER ****
'WM30>59' M61 WM30=0                            ;Timer based Transmit (M61)
'M61=1 & !(RO37=1)' RO38=8 RO37=1 ;                             ; RO37 & RO38 are special FTP sending registers, look
manual!

;Backup SMS sending if enabled
;Send if error in FTP transfer. Exclude file empty error (130)
;'RO37>2&!(RO37=130)' "WM20,%WM21,%WM22,%WM23,%WM100,%RO98" 20 ; sends SMS to phone number 20 defined in phone book!

; Note when testing this application use direct cable connection and DIP 4 position = ON, before testing the real
GPRS transfer.
; Look incoming files in alarm log view and the GsmProgrammer program root directory.
```

12.5 Configuring DB2SMS

Open *config.ini* file that is located in the installation folder of DB2SMS. By default the file includes ConnectionString-parameter that is suitable to be used with PostgreSQL. Server, user name and password should be located in the connection string (example: *ConnectionString='DRIVER={PostgreSQL UNICODE};SERVER=localhost;DATABASE=smsindusoft;Uid=iws;Pwd=passwd;'*).

12.6 What to do next

If you are using SMS data you should next read:
DB2SMS_UM_v102_eng.pdf
DB2SMS_Using_with_WebStudio_UM_v101_eng.pdf

If you are using FTP data you should next read:
FileParser_GPRS_FTP_driver_for_Indusoft_UM_v11.pdf

After that you can learn more about AutoLog GSM-PLC application programming with GSMProgrammer:

AutoLog_GSM_PLC_Service_Manual.pdf

Then you could learn more about Indusoft Web Studio SCADA by reading:

Getting_Started_Guide_v70.pdf
Technical_Reference.pdf

AND the most important!

Try creating your own application, using DEMO applications as template or from the scratch!

FF-Automation can help you to get started by arranging technical training and project consultancy.